OPIRA: The Optical-flow Perspective Invariant Registration Augmentation and other improvements for Natural Feature Registration

---

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy
in the
University of Canterbury
by
Adrian Clark

University of Canterbury
2009

Publications

Material from this thesis has been previously published in the peer-reviewed papers listed below. The sections of the thesis the publications relate to are noted. The following papers are predominantly my own work:

1. Clark, A. and Green, R. (2005) "An Adaptive Algorithm Switching System for Image Based Object Registration". Image and Vision Computing New Zealand, November 2005. (Section 9.2.2)

2. Clark, A. and Green, R. (2006) "Detection and Removal of Global and Local Noise in Realtime Video Streams". Image and Vision Computing New Zealand, November 2006. (Section 6.1)

3. Clark, A. and Green, R. and Grant, R. (2007) "Image and video noise - a comparison of noise in images and video with regards to detection and removal". VISAPP 2007: Proceedings of the Second International Conference on Computer Vision Theory and Applications, Barcelona, Spain, March 8-11, 2007. (Sections 6.1-6.2)

4. Clark, A. and Green, R. and Grant, R. (2008) "Perspective correction for improved visual registration using natural features". Image and Vision Computing New Zealand, November 2008. (Chapter 5).

The follow papers were co-authored by me, and have information relevant to this work:

1. Grant, R.N. and Green, R.D. and Clark, A.J. (2007) "Hue variance prediction - an empirical estimate of the variance within the hue of an image". VISAPP 2007: Proceedings of the Second International Conference on Computer Vision Theory and Applications, Barcelona, Spain, March 8-11, 2007. (Section 6.1.1)

2. Grant, R.N. and Green, R.D. and Clark, A.J. (2008) "HLS Distorted colour model for enhanced colour image segmentation". Image and Vision Computing New Zealand, November 2008. (Section 2.1.2)

Dedicated to
Lynne and Robert

Abstract

In the domain of computer vision, registration is the process of calculating the transformation between a known object, called a marker, and a camera which is viewing it. Registration is the foundation for a number of applications across a range of disciplines such as augmented reality, medical imaging and robotic navigation.

In the set of two dimensional planar markers, there are two classes: (1) fiducial, which are designed to be easily recognisable by computers but have little to no semantic meaning to people, and (2) natural features, which have meaning to people, but can still be registered by a computer. As computers become more powerful, natural feature markers are increasingly the more popular choice; however there are still a number of inherent problems with this class of markers.

This thesis examines the most common shortcomings of natural feature markers, and proposes and evaluates solutions to these weaknesses. The work starts with a review of the existing planar registration approaches, both fiducial and natural features, with a focus on the strengths and weaknesses of each. From this review, the theory behind planar registration is discussed, from the different coordinate systems and transformations, to the computation of the registration transformation.

With a foundation of planar registration, natural feature registration is decomposed into its main stages, and each stage is described in detail. This leads into a discussion of the complete natural feature registration pipeline, highlighting common issues encountered at each step, and discussing the possible solutions for each issue.

A new implementation of natural feature registration called the Optical-flow Perspective Invariant Registration Augmentation (OPIRA) is proposed, which provides vast improvements in robustness to perspective, rotation and changes in scale to popular registration algorithms such as SIFT, SURF, and the Ferns classifier. OPIRA is shown to improve perspective invariance on average by 15% for SIFT, 25% for SURF and 20% for the Ferns Classifier, as well as provide complete rotation invariance for the rotation dependent implementations of these algorithms.

From the investigation into problems and potential resolutions at each stage during registration, each proposed solution is evaluated empirically against an external ground truth. The results are discussed and a conclusion on the improvements gained by each proposed solution and the feasibility of use in a real natural feature registration application is drawn.

Finally, some applications which use the research contained within this thesis are described, as well as some future directions for the research.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am indebted to many people for their help in making this thesis possible. At this stage I wish to acknowledge and thank them for their support.

First and foremost, I must thank my supervisor Dr Richard Green for his constant support, encouragement and guidance throughout this work. Richard has always been positive and encouraging, and his enthusiasm for his students is unparalleled. I feel honoured to have had him as a mentor.

Secondly I would like to thank my co-supervisor Dr Mark Billinghurst. Mark's dedication to his work and the students at the HIT Lab NZ is an inspiration, and I am proud to have had the opportunity to study in the environment he has created.

I would like to thank the examiners of this thesis for taking the time to read and evaluate my work.

I cannot express enough gratitude to my family, who have always supported and believed in me. To my parents, Lynne and Robert, my siblings Nathan, Tahlia and Jeremy, and the Ono family, thank you so much for the encouragement, patience and love.

I would not have been able to complete this thesis without the support of my friends, who always kept me positive and had time to listen. Thank you Krystal, Kyle, Andrew and Paul. I would like to say a special thank you to the Grant family; Tony, Judy and Nikki, you have always made me feel like part of your family. Last but not least, thanks to baby Lance, you always bring a smile to my face.

Finally I would like to thank the staff and students at the HIT Lab NZ, especially Raphaël, Julian and Robert. Your assistance and advice has been invaluable, and I cannot express enough gratitude for the time you have invested in helping me during this research.

Chapter 1

Introduction

1.1 Thesis Overview

Computer vision is a term which encompasses a large domain of research. Generally speaking, computer vision is concerned with computations performed on digital images, such as those captured by digital or video cameras. One important aspect of computer vision is registration.

Registration is the process of calculating the transformation between a known object, called a marker, and the camera which is viewing it. This transformation is a representation of the location and orientation of the marker from the perspective of the camera in three dimensional space. The orientation is comprised of three rotations; yaw, pitch and roll, and the location is three translations; vertical, horizontal, and in depth.

As shown in Figure 1.1, registration provides the foundation for a wide range of different types of computer vision based applications:

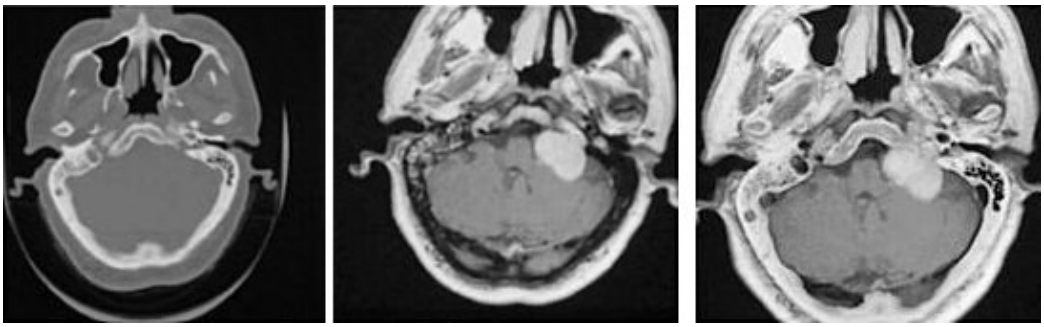
- Augmented Reality (AR) requires the location of the object to augment the real world with computer graphics.

- Robotics platforms use registration to identify and navigate the environment they are operating in.

- Medical imaging systems register acquired images to correlate the data contained in the image to a known model.

- Video compression attempts to find areas of commonality between frames in order to minimise the amount of new information which needs to be stored in each frame.

(a) (b)



(c)

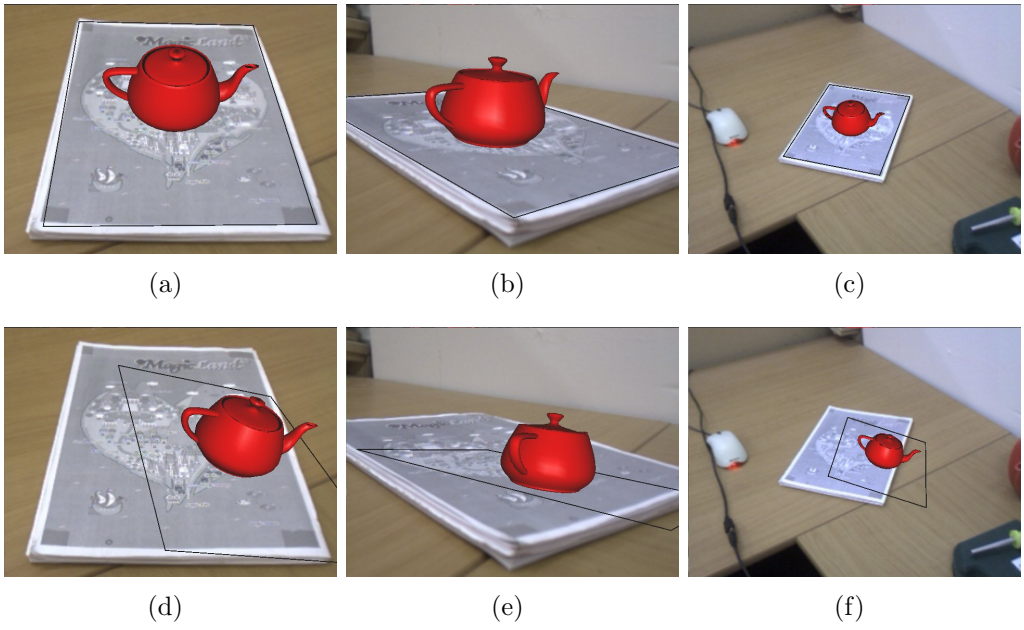
Figure 1.1: Some different uses of registration, (a) Augmented reality - the MagicBook (Billinghurst et al. 2001), (b) Robotics - location and mapping (Tomono 2007), (c) Medical imaging - correlation of CT and MR data (Hajnal et al. 2001)

Registration can be further subdivided into two main subsets, planar registration and 3D model based registration. The focus of this research is planar registration, where the registration marker is a two dimensional surface.

There are two major classes of markers used in planar registration. The first, and simplest, is the fiducial marker, which is specially designed for easy and fast recognition by computer vision algorithms, but has little or no semantic meaning for people. The second type of planar marker, and the focus of this research, is the natural feature marker. A planar natural feature marker is any two dimensional surface which can be registered using computer vision, while still carrying semantic meaning. This means that any existing two dimensional surface with sufficient detail can be used as a

natural feature marker without needing modification.

The accuracy of registration is paramount to the success of any application which utilises it. Figure 1.2 shows an example of good registration (top) and poor registration (bottom) in the context of an augmented reality application. With good registration, the computer generated teapot is rendered on the marker creating the illusion that the teapot is physically present in 3D space, and attached to the marker. In the case of poor registration, the teapot moves around independently and the user experience is diminished.



(a) (b) (c)

(d) (e) (f)

Figure 1.2: The difference between accurate and inaccurate registration, the teapot appears to be sitting on the marker (top), the teapot is floating around space (bottom)

In this research, the common causes of failure and inaccuracies in natural feature registration are examined, and solutions are proposed and evaluated. A new implementation of natural feature registration called Optical-flow Perspective Invariant Registration Augmentation (OPIRA) is proposed and evaluated. OPIRA improves robustness to perspective, rotation and changes in scale. The improvement in registration as a result of each solution is determined, and the feasibility of each solution in a natural feature registration

application is discussed.

1.2 Chapter Summary

This thesis was written with a bottom-up perspective of natural feature registration, beginning with issues with planar registration in general, and gradually expanding to include problems specific to natural feature registration and the natural feature registration pipeline. A summary of each chapter follows.

Chapter 2: Background Research examines a range of well known planar fiducial and natural feature registration algorithms with a focus on the strengths and weaknesses of each algorithm.

Chapter 3: Planar Registration describes the theory behind planar registration. Computer vision coordinate systems and a well-known camera calibration algorithm are discussed, in the context of calculating the registration transformation.

Chapter 4: Natural Feature Registration describes how natural feature registration algorithms provide the necessary information for registration computation. Natural feature registration is decomposed into its core components, with a focus on the potential weaknesses of each component.

Chapter 5: OPIRA: The Optical-flow Perspective Invariant Registration Augmentation discusses how standard natural feature registration is performed, and introduced optical tracking for registration. A new registration method called OPIRA is proposed which reduces the effect of changes in scale, rotation, and perspective distortion for natural feature registration.

Chapter 6: Other Improvements for Natural Feature Registration discusses the weaknesses identified in Chapter 4 which are not resolved by OPIRA, and suggests methods of reducing their impact.

Chapter 7: Evaluation provides an empirical evaluation for the solutions discussed in the previous chapters.

Chapter 8: Discussion of Results evaluates the results found in the previous chapter and discusses the feasibility of using each solution in a natural feature registration application.

Chapter 9: Applications and Future Work describes some of the applications which have come as a result to the research contained in this thesis, and future directions for the research.

Chapter 10: Conclusion provides a concise summary of the results obtained in the thesis, and how natural feature registration can be improved using the techniques examined.

1.3 Research Contributions

The main contributions of this thesis are:

- A critical review of existing planar registration algorithms, including popular fiducial and natural feature based algorithms.

- A comprehensive discussion of the theory of camera based coordinate systems, camera calibration, natural feature registration and optical tracking as a detailed analysis of the registration process.

- A study of the different steps of the registration process, leading to the proposal of methods of improvement and the introduction of a new algorithm called Optical-flow Perspective Invariant Registration Augmentation (OPIRA) which improves perspective, scale and rotation robustness for all two dimensional planar visual registration systems. The algorithm is formally evaluated and shows significant improvements over the most popular registration algorithms.

- A detailed and systematic evaluation of the proposed improvements for natural feature registration algorithms, complemented by an analysis

of the improvements of OPIRA. From these findings recommendations were proposed regarding development of natural feature registration systems.

- A proposed framework for minimising image transformations and distortions which affect the performance of natural feature registration. Methods to minimizing these effects are proposed and formally evaluated to determine their efficacy, and the benefits and drawbacks of each method discussed.

- A software framework based around OPIRA for designing, testing and deploying natural feature registration algorithms. The framework is sufficiently robust to be used for augmented reality or robotic navigation.

Chapter 2

Background Research

In this chapter common planar registration algorithms are discussed with a focus on each algorithm's strengths and weaknesses. Planar registration is the process of calculating the position and orientation of a two dimensional object, known as a marker, from the perspective of the camera viewing it. There are two classes of markers in planar registration; those specifically designed for registration called fiducial markers, and those not specifically designed for registration but which have a sufficient level of detail to allow pose computation, called natural feature markers.

The following sections discuss common techniques for both fiducial and natural feature registration.

2.1 Fiducial Markers

The term fiducial is used across a range of contexts to describe a point of reference. In computer vision based registration, a fiducial is a marker designed specifically for registration. A fiducial marker may be a single planar object or may be comprised of multiple objects arranged in a defined pattern. Although studies of fiducial markers have been conducted ((Fiala 2005b), (B., Xaio and Middlin 2002)), the optimal design of fiducial markers depends on their application. Figure 2.1 shows a range of different fiducial markers (Fiala 2005b).

Fiducial markers are usually designed to maximise the robustness of registration, while minimising computational load for optimal performance. Robustness is achieved by designing the marker to be highly visible and easily identifiable to minimise the impact of poor or variable lighting, any change in perspective relative to the camera, and other confounding factors typically encountered in the proposed application. The performance of the registra-

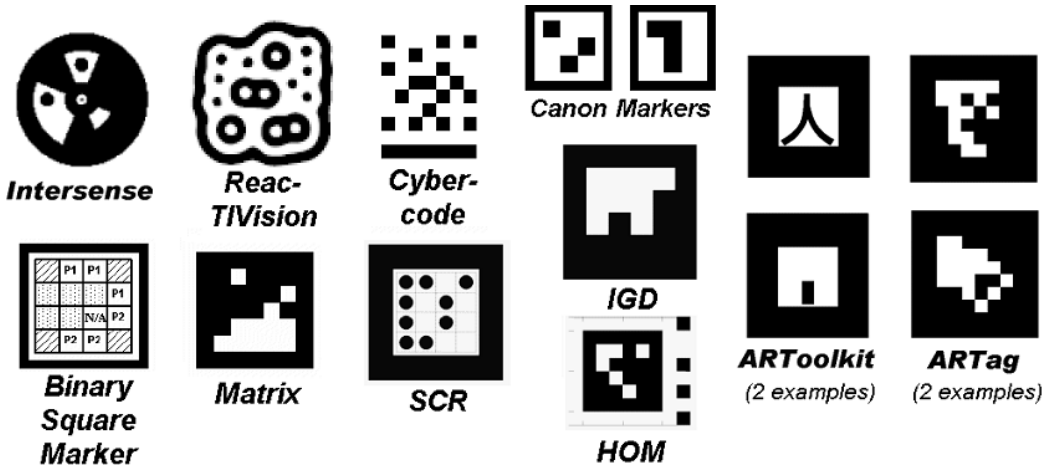


Figure 2.1: A selection of fiducial markers (Fiala 2005b)

tion algorithm can be improved by designing the marker such that it can be located and the pose calculated with very simple and efficient vision algorithms, freeing up processing time for other application specific tasks.

The disadvantages of fiducial markers become obvious in applications which require user interaction, particularly with untrained users. The markers are often large and distracting, particularly in augmented reality applications where the illusion of virtual media existing in the real world is degraded by the use of a marker which has no semantic meaning to the user. This lack of semantic meaning also limits the versatility of the medium, for example fiducial markers for augmented reality advertisements in printed media take up valuable and often expensive space in the publication, and provide no information for readers who are unable or unwilling to view the digital content.

Fiducial markers are often prone to failure under occlusion unless redundancy is considered when the marker is designed. For certain markers registration can fail if even the smallest part of the marker is occluded. This presents a major problem when the application requires user interaction with the marker; the user must change their behaviour to ensure they do not occlude the marker, which can be difficult if the perspective of the camera is

not similar to the users own perspective.

Despite these downfalls, Lepetit and Fua (2005) state that "practical vision-based 3D tracking systems still rely on fiducials because this remains the only approach that is sufficiently fast, robust, and accurate". Several popular fiducial registration algorithms are discussed in the following sections.

2.1.1 Light Beacons

Fiducial markers can be as simple as using one or many distinct light beacons somewhere in the operating space (Bajura and Neumann 1995). Fiducials which emit light have the benefit of being robust to poor or variable lighting in the application environment. The beacons are invariant to most changes in the colour and intensity of any light sources and the effect of shadows, and are the only marker which can be registered in an environment with no ambient light.

Light beacons appear in the captured image as points of high intensity, which can be located with a computationally efficient threshold of colour or brightness. To form a correct registration transformation matrix, four unique points are required, as explained in Section 3.3. To identify orientation of the lights for rotation invariance, each one may be a different colour, or turned off and on in sequence (Azuma and Bishop 1994). If visible light beacons are detrimental to the user experience, infrared beacons can be used (Welch, Bishop, Vicci, Brumback, Keller and Colucci 1999).

To use a single beacon for registration, information can be encoded in a single beacon using a special patterned lens over the light beacon, known as a Bokode marker (Mohan, Woo, Hiura, Smithwick and Raskar 2009). This information is only visible to a defocused camera, where the information stored in the point of light is expanded to a disk on the image plane. This information may be viewed up to a distance of several metres from the object. Figure 2.2 shows an example of this technique.

Despite their advantages, light beacons are unsuitable for many applications. The marker may be large and cumbersome, and require a power source. A partial solution to this is the use of markers constructed from a

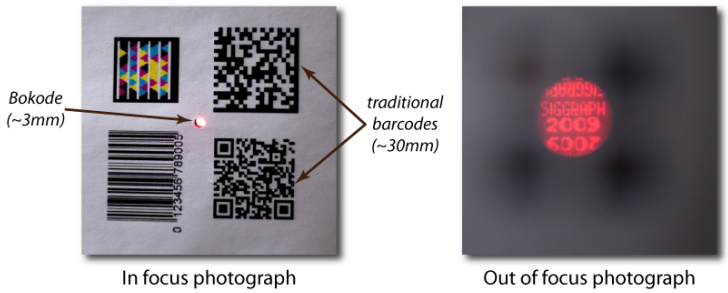


Figure 2.2: A Bokode fiducial marker, which conceals information using a light beacon with a special lens which can only be viewed with a defocused camera (Mohan et al. 2009)

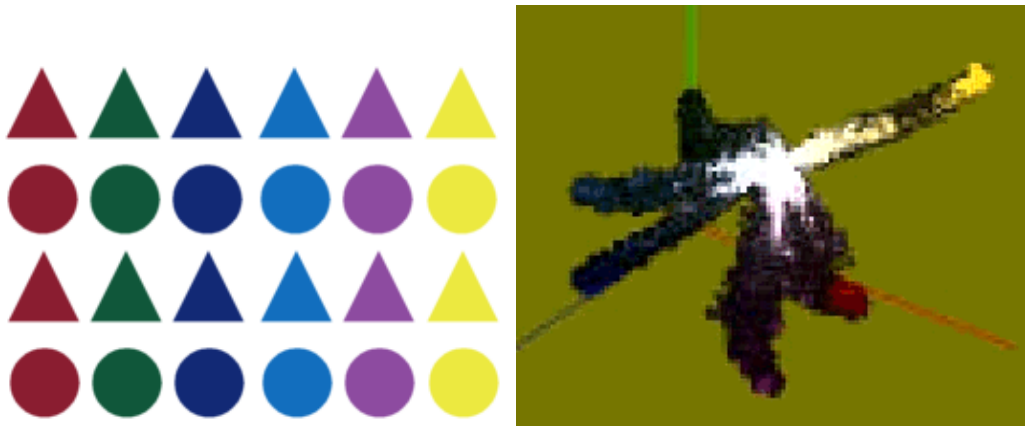
retro-reflective material, and mounting the illumination source on the camera (Ribo, Pinz and Fuhrmann 2001). Care must also be taken to ensure that the intensity of the light does not affect the sensor in the camera, otherwise the overall quality of the image may be degraded.

2.1.2 Fiducial Features

A fiducial feature marker is a single marker which is comprised of multiple small distinct fiducials, or features. These features are in a known pattern which is used to determine the identity, position and orientation of the marker. Fiducial feature markers lack the invariance to lighting effects of light beacons, but are smaller and easier to integrate into an existing environment, and require no power.

The features are typically small shapes of a uniform colour (Koller, Klinker, Rose, Breen, Whitaker and Tuceryan 1997), which can be located with a computationally efficient hue filter in order to compute a registration matrix. Figure 2.3 shows the fiducials used by Cho, Park and Neumann (1997), and each fiducial's position in the colour cube.

The ease of adding additional features makes fiducial feature markers suitable for registration in large environments (Neumann and Park 1998). Additional fiducial features can be added to the scene in real-time and their positions in the environment estimated to dynamically increase the physical size of the total registration area (Park, You and Neumann 1998). Scale



(a) (b)

Figure 2.3: Fiducial features used by Cho et al. (1997), (a) The fiducials, (b) Their distribution in the RGB colour cube

invariance can be achieved by using features of different sizes; in this way smaller features will only be visible when the camera is close to the scene and larger features visible when the camera is further away. Another solution was proposed by Cho, Lee and Neumann (1998), using multi-ring fiducials, as shown in Figure 2.4. Each ring is visible at a different distance.

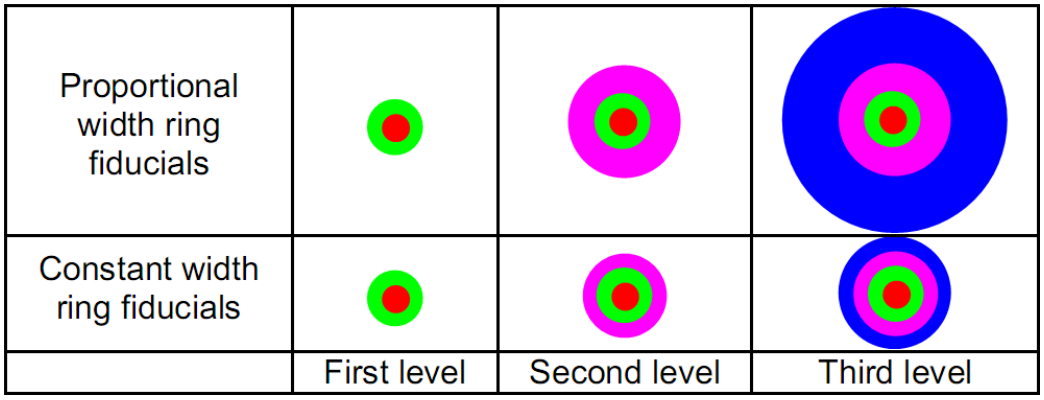
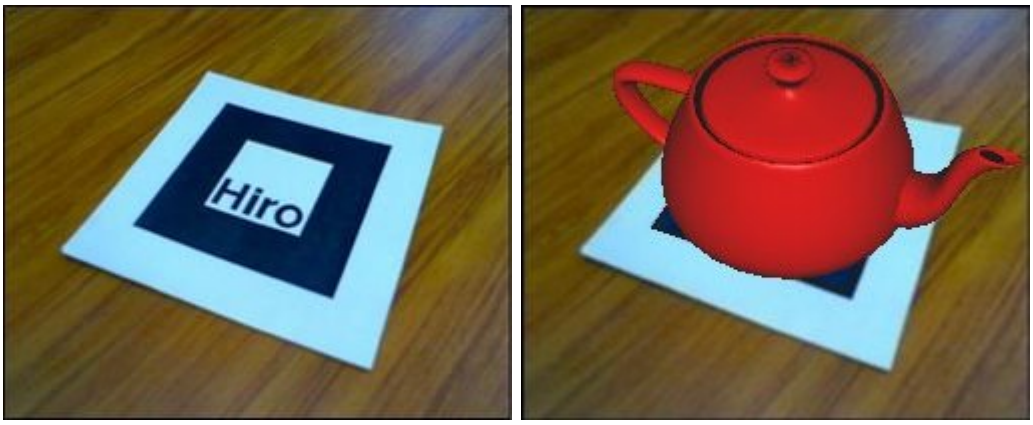


Figure 2.4: Multi-ring fiducials used by Fiducial features used by Cho et al. (1998), in the top images the ring widths are proportional to the ring level, in the bottom images the ring widths are constant.

The accuracy of registering a fiducial feature marker depends on the visibility and differentiability of the features chosen. Features which are too small are more impacted by noise, while larger features are prone to occlusion and require the camera to be further from the scene to ensure enough features are in frame to perform registration. The robustness of coloured fiducials will depend on the lighting of the environment, and the colour model used to describe them (Grant, Green and Clark 2008).

2.1.3 ARToolKit

The ARToolKit software library (Kato and Billinghurst 1999) is a widely known and used fiducial marker based registration system. ARToolKit uses visual tags based on 2D bar codes, first made popular by Rekimoto and Ayatsuka (2000). In ARToolKit, these visual tags are unique identifying symbols contained within a black frame, as shown in Figure 2.5.



(a) (b)

Figure 2.5: An example marker used by the ARToolKit, and the marker with a teapot rendered on it using the ARToolKit

The ARToolKit library performs registration by partial line fitting on a binary thresholded image. Any regions enclosed by four line segments are considered potential markers. Using the perspective projection matrix of the camera, which is calculated in an off-line calibration stage, the corresponding planes for each of the line segments are calculated. The unit direction vector

of each pair of parallel line segments is found by calculating the outer product of the normals of the plane formed by the line segments. When the unit direction vector is found for two sets of parallel lines enclosing a region, the rotation component of the transformation matrix is generated from those two direction vectors, with the normal vector forming the Z axis. Once the rotation component of the transformation matrix is found, the translation component is calculated from the coordinates of the markers vertices. An iterative sum of difference approach is used to reduce error by finding the difference between the location of the vertices in the image frame and the locations specified by the transformation. Once the transformation is found, virtual content can then be overlaid, as shown in Figure 2.6.

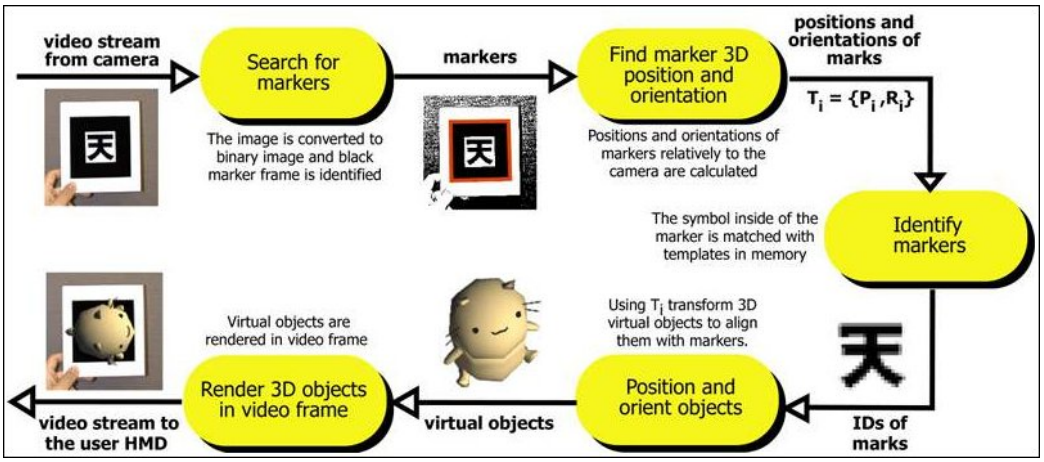


Figure 2.6: The ARToolKit registration process[1]

The registration method described is not rotation invariant, as a single square marker has four-fold rotational symmetry. This is resolved by having a symbol contained in the marker with no axes of rotational symmetry. To calculate the rotation of the marker, the symbol inside the square is normalized and the four possible orientations generated. The four rotations of the symbol are matched to a database of symbols known to the system, the orientation with highest match correlation to a known symbol is assumed to

[1] `http://www.hitl.washington.edu/artoolkit/documentation/userarwork.htm`

be the correct rotation. This matching process also provides the means of differentiating between markers.

Multiple ARToolKit markers can be used to create a projective space (Uematsu and Saito 2005). As each ARToolKit marker can be used to calculate a coordinate space, any arbitrary space can be created using multiple markers to define the boundary. This can even be done dynamically by calculating the transformation from the projective matrix of each marker to the projective space. This allows arbitrary extension of the tracking environment, in a similar way as the fiducial feature markers in Section 2.1.2, and robustness to occlusion, as long as at least one marker is visible.

ARToolKit has been used in a range of applications such as remote teleconferencing (Billinghurst, Cheok, Prince and Kato 2002), wide area applications (Wagner 2002) and hand position tracking (Piekarski and Thomas 2002). ARToolKit has been ported to work on mobile phones (Henrysson, Ollila and Billinghurst 2005) and the PocketPC (Wagner and Schmalstieg 2003). Juan, Joele, Botella, Baños, Alcañiz and van der Mast (2006) presented a system using "invisible" ARToolKit markers, which were created using a special ink which cannot be seen by the human eye, but are visible to a camera with an infra red filter.

In addition to the problems inherent in all fiducial marker registration algorithms, the line fitting algorithm used by ARToolKit for marker detection is highly susceptible to occlusion; even a minor occlusion of the edge of the marker will cause failure of the region detector. When using multiple markers, the success of registration is highly influenced by the amount of difference between the patterns used for each marker, markers with similar characteristics are often mistaken for one another using the template matching approach.

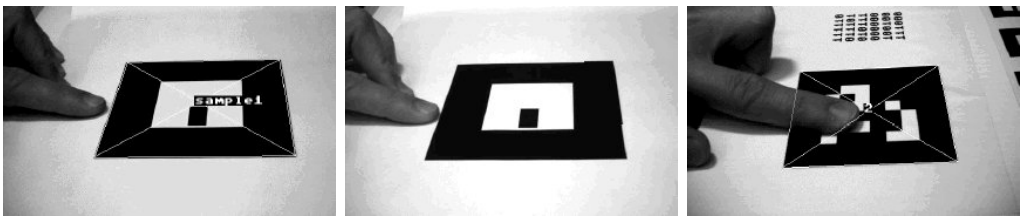
2.1.4 ARTag

ARTag is a successor to ARToolKit which was designed to address a number of the short comings, such as susceptibility to occlusion and mis-registration due to template matching (Fiala 2005a). An edge-based algorithm designed to find quadrilaterals with heuristics of line segments is used to allow for

breaks in the continuity of lines defining the edges of the marker, while also improving robustness to poor and variant lighting.

Instead of a symbol inside the marker for identification and orientation, a six by six binary grid is used for identification, using a similar idea as a bar code. The code has added redundancy in the form of a checksum to reduce the likelihood of false positives during marker matching. If the checksum does not match due to noise in the image or occlusion of part of the marker, the system is capable of calculating the most probable candidate by finding the minimum distance with other markers in the database.

Figure 2.7 shows the occlusion invariance of the ARTag system compared to ARToolKit.



(a) (b) (c)

Figure 2.7: ARTag occlusion invariance compared with ARToolKit, (a) ARToolKit working without occlusion, (b) ARToolKit fails with even slight occlusion of the marker boundary, (c) ARTag successfully registers despite large occlusions (Fiala 2004)

The 11 bit ARTag code allows for 2001 markers (46 markers are invalid due to rotational symmetry). To minimise the occurrence of mismatches, the markers are assigned in order of maximum distance between the new marker and existing markers. Figure 2.8 shows the first six markers in sequential order.

The symbols used by ARToolKit can be used to represent the purpose of the marker, while the bar code system used in ARTag has no semantic meaning, and are not human readable.

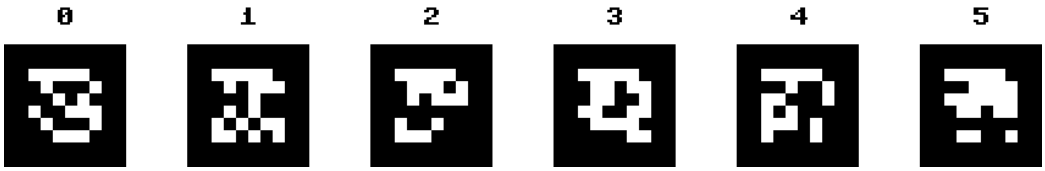


Figure 2.8: The first six markers of the ARTag system, chosen due to maximum difference between the codes (Fiala 2004).

2.1.5 ARToolKitPlus and Studierstube ES

ARToolKitPlus (Wagner and Schmalstieg 2007) optimised the concepts presented in the ARToolKit and ARTag systems. This allowed for complex augmented reality applications to run in real time, even on mobile devices. In addition to system wide improvements to maximize the computational performance of the system, problems specific to cameras on mobile devices, such as vignetting, were addressed to improve the accuracy of the system. The result was a fiducial marker based augmented reality system capable of running in real time on mobile devices.

(a) (b)

Figure 2.9: The frame markers from Studierstube tracker, which contain all the information required for registration in the border[3]

[3] `http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php`

The ARToolKitPlus was later rewritten from scratch and renamed the Studierstube Tracker. This tracker is part of the Studierstube ES Framework (Schmalstieg and Wagner 2007) for augmented reality on hand held devices. The Studierstube tracker added some new features such as frame markers, shown in Figure 2.9, which encode all the data required for registration into a frame so that any content can be placed inside the frame. They also introduced ISO standard data matrix markers which can contain digital information within the marker itself.

2.2 Natural Feature Markers

Natural features are structures within an object which have semantic meaning and can be identified regardless of changes in viewpoint. In two dimensional planar registration any flat surface with a significant number of visible natural features can be used as a natural feature marker without the need for modification. In an augmented reality application natural feature markers can contain contextual meaning with the associated virtual content. An example of this is the "Giant Jimmy Jones" MagicBook (McKenzie and Darnell 2003), which appears to be an ordinary story book, but with the augmented reality hardware the story becomes animated in three dimensions on the pages, as shown in Figure 2.10.

Natural feature registration uses algorithms which examine image structures across multiple dimensions. Structures which can be found in the marker regardless of viewpoint, known as feature points, are found either in zero dimensions, such as points or corners, or in one dimension, such as lines and edges (Neumann and You 1999). In order to differentiate between these features points, the surrounding area is examined in two dimensions. These areas are algorithmically converted into a unique descriptor for the feature, which can then be compared to other feature descriptors to evaluate matches. The algorithms can be as simple as an intensity change (Jurie and Dhome 2002), or more complicated such as the SIFT descriptor. These multi-dimensional features can be combined with other image features such as contours to increase the information available to the registration algorithm (Masso, Dhome and Jurie 2003).

Figure 2.10: The Giant Jimmy Jones MagicBook (McKenzie and Darnell 2003)

The process of natural feature registration involves finding a set of natural features within the marker, a set of features from the input image being registered, and then searching for matches between the two sets. With a minimum of four matching features, a homography can be computed and the registration transformation computed.

The following sections describe common natural feature registration algorithms.

2.2.1 ARToolKit NFT

A new version of the ARToolKit was developed which combines the existing fiducial registration algorithm with a texture based tracking algorithm, called ARToolKit NFT (Kato, Tachibana, Billinghurst and Grafe 2003). Initial registration of the marker is performed using an ARToolKit marker, and this initial transformation of the object is used to initialise the natural feature tracking algorithm, which is used for all consecutive registration computations. Figure 2.11 has an example of an ARToolKit NFT marker.

After initial registration of the ARToolKit marker has succeeded, the

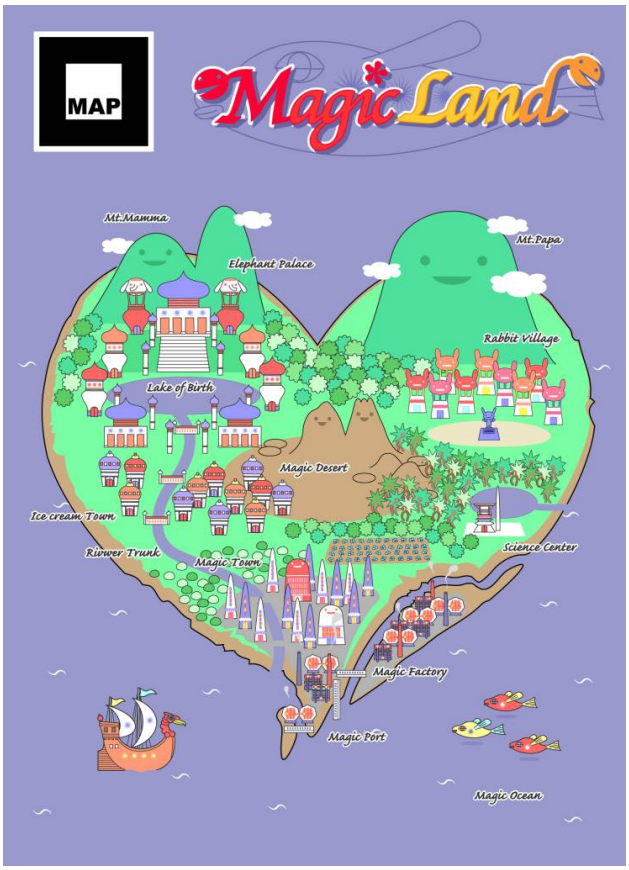


Figure 2.11: The "MagicLand" marker designed to work with the ARToolKit NFT

transformation matrix is used to calculate the position of known feature points in the image. For each successive image, these coordinates are transformed by the velocity of the image over previous frames to estimate the new positions of the features. Template matching using normalized cross-correlation is performed on the estimated coordinates to obtain an exact match for the new positions of the features. The normalized cross-correlation equation is shown in Equation 2.1:

$$s = \frac{\sum_{i=1}^{N}(x_i - \tilde{x}) \cdot (y_i - \tilde{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \tilde{x})^2}\sqrt{\sum_{j=1}^{N}(y_j - \tilde{y})^2}} \tag{2.1}$$

where x_i is the pixel value, $\tilde{x}$ is the mean of pixel values, y_i is the template value and $\tilde{y}$ is the mean of template values. The template size used is 24×12 pixels for a 640×480 image. The search window which is examined is 49×49 pixels. The output s will be a value from -1.0 to 1.0, where a larger number represents a closer match between the window and the template. Experimentally a threshold of 0.7 was chosen as the minimum correlation required for a successful match.

ARToolKit NFT employs an off-line training phase to select repeatable features, which are features in the marker which can be found regardless of changes in viewpoint. Multi-scale templates are taken to add scale invariance of the features detected, and a 3×3 averaging filter is applied to remove high frequency features which are unlikely to be detected by a camera which does not have a high resolution sensor or does not focus light perfectly. Features which have no similar regions within the search window, but a high similarity within a 5×5 pixel window around the feature point, are chosen provided they do not overlap with existing feature templates. This off-line training phase is extremely slow, taking hours to train a book of twenty A4 sized pages at 300dpi on a standard desktop computer.

To stabilise tracking while increasing speed, the system attempts to match only four features. These are selected in a specific order:

1. The feature point furthest from the image centre

2. The feature point furthest from the first feature point

3. The feature point which makes the triangle with the largest area with the first and second feature points

4. The feature point which makes the rectangle with the largest area with the first, second and third feature points

If the error of the observed screen coordinates compared to the transformed object coordinates is too high, a fifth feature will be selected based on the order the features were found. If there is no fifth feature point visible,

all combinations of three features points from the four found will be tested, and the three which provided the lowest error will be chosen.

The need for a fiducial marker for initial registration means that existing objects cannot be used as markers without first making modifications to them, which may be impracticable or impossible. Before a successful initial registration, ARToolKit NFT suffers from all the same weaknesses as the ARToolKit, such as high susceptibility to occlusion of the marker.

After the initial registration, ARToolKit NFT system overcomes the shortcomings of the ARToolKit. It is more robust to occlusion, as the feature points provide a greater registration area, and once initial registration has occurred, the ARToolKit marker no longer needs to remain in the camera's view. The decision to attempt registration on only four features to increase speed vastly decreases the accuracy of the registration transformation obtained, where a small transformation of a single feature due to noise or inaccuracy in image capture will result in large changes in the final transformation. This is particularly noticeable when the marker approaches large perspective transformations, as the relative distance between pixels increases. This results in quite severe "jittering" of any digital content which is overlaid on the marker.

The most recent implementation of the ARToolKit NFT no longer requires an ARToolKit marker for initial pose estimation, but instead uses colour based template matching (Taketa, Hayashi, Kato and Nishida 2007). Multiple markers are handled by a page determination algorithm which is trained on markers at $\frac{1}{16}$ resolution.

2.2.2 Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform (SIFT) is a patented registration algorithm developed by Lowe (1999) and inspired by the function of neurons in the inferior temporal cortex in primates. Difference of Gaussian (DoG) scale spaces are computed for the image across a number of image sizes, or octaves, as shown in Figure 2.12.

The scale invariant features are the local maxima and minima in the DoG which have been located by comparing each pixel against its eight neighbours

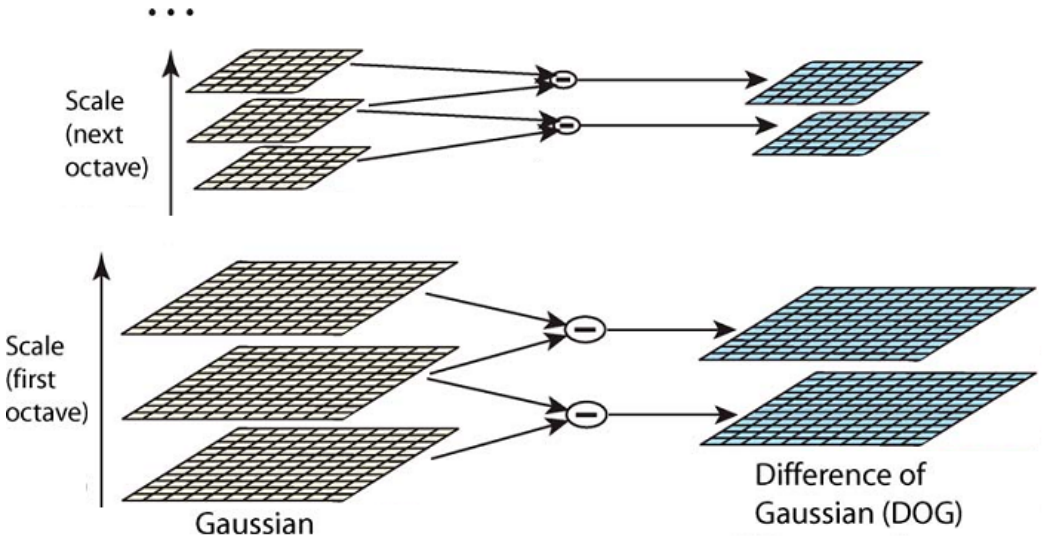


Figure 2.12: The DoG computation for SIFT. The original image is incrementally convolved with the Gaussian operation (black grids on left), and the difference between sequential operators is found (blue grids on right). This is done across a number of octave scales, here two are shown, one half the scale of the other (Lowe 2004)

in the current scale, and the eighteen neighbours in adjacent scales, as shown in Figure 2.13. A feature vector is calculated relative to its scale-space coordinate frame. These vectors are assigned an orientation based on the peak in a histogram of local image gradient orientations to allow rotation invariance (Lowe 2004).

Once a feature has a stable location, scale and orientation, a local descriptor is calculated which allows for identification and differentiation. These descriptors are modelled on the function of complex neurons in the visual cortex (Edelman, Intrator and Poggio 1997), with the goal of providing invariance to noise, minor changes in 3D projection, and some robustness to small changes in content. Image gradient magnitudes and orientations from 16×16 points around the feature are sampled and rotated relative to the feature orientation. Each point is weighted using a Gaussian function to give less emphasis to points further away from the feature. These points are compressed down into a 4×4 sample region of 8 directional orientation

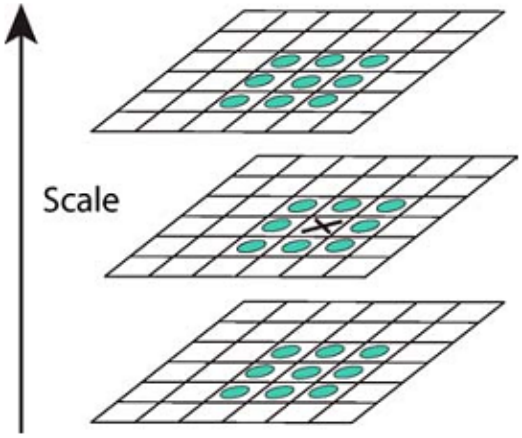


Figure 2.13: Maxima and minima are detected by comparing a pixel with the eight neighbouring pixels in the current scale, and the eighteen neighbours in adjacent scales (Lowe 2004)

histograms, which count the number of occurrences of each gradient orientation, such that a gradient sample can shift up to 4 sample positions before the descriptor is not longer equivalent. The descriptor is a vector which contains the values of all the orientation histograms, resulting in a 128 value descriptor (8 orientations$\times$(4$\times$4) array). The descriptor vector is normalised to unit length to remove illumination dependence. Features are matched by comparing the descriptors using a nearest neighbour approach such as Best-Bin-First (Beis and Lowe 1997).

The SIFT algorithm is not without limitations. The processing time taken for SIFT is considerable, making it unsuitable for real-time applications such as augmented reality. Fast Approximated SIFT was designed to improve the speed of SIFT, by using a Difference of Mean integral approximation of DoG, and integral histograms instead of orientation histograms to "(increase speed) by a factor of eight, while the matching and repeatability performance in decreased only slightly"(Grabner, Grabner and Bischof 2006). PCA-SIFT applies principal component analysis on the gradient image, which can reduce the descriptor length to 36, allowing for a faster matching time (Ke and Sukthankar 2004). Unfortunately PCA-SIFT was proven to be less distinctive than SIFT (Mikolajczyk and Schmid 2005), and the additional processing

required for principal component analysis reduces the overall speed gains.

To improve the robustness of SIFT, Lemuz-López and Arias-Estrada (2006) combined the SIFT algorithm with the iterative closest point scheme used in medical imaging. ICP-SIFT has a higher ratio of correct to false positive matches than SIFT alone, with an RMS error of less than half that of SIFT on the evaluated datasets. The trade off for this improved performance is the increased processing time due to the iterative nature of the algorithm.

Mikolajczyk and Schmid (2005) found that repeatability, a measure of how many features were detected in both the source and registration images in relation to the lowest total number of feature points found, of the SIFT feature points decreases as the viewpoint angle to the image plane increases, with only 50% repeatability at 50 degrees rotation.

2.2.3 Speeded Up Robust Features (SURF)

Speeded Up Robust Features or SURF (Bay, Tuytelaars and Van Gool 2006), is a rotational and scale invariant feature point detector and descriptor. Designed to outperform other feature detectors in speed, the algorithm approximates more robust detectors such as SIFT, providing a comparable level of accuracy while minimising computational load. The SURF descriptor is robust to changes in scale and rotation, while perspective distortion and other "second-order effects" are not explicitly handled.

The SURF feature detector operates on integral images to maximise speed. Every pixel (x, y) in an integral image $I_{\sum}$ is the sum of all pixels in the original image I in the rectangle formed between the image origin and pixel (x, y) (Viola and Jones 2001). This process is defined by Equation 2.2.

$$I_{\sum}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{2.2}$$

The SURF feature detector uses an approximation of the Hessian matrix on the integral images called the Determinant of Hessian (DOH), which provides good accuracy while still being fast to compute. The Gaussian second order derivatives required for computation of the Hessian matrix are approximated using rectangular integer averages known as box filters, which are

very fast to compute using the integral images.

Rotation invariance is achieved by identifying a reproducible orientation for each feature point. A circular window is defined at six times the scale at which the feature point was detected, and the Haar wavelet responses inside this window are calculated. The wavelet responses are represented as vectors and the dominant orientation is calculated from the sum of responses in a $\frac{\pi}{3}$ sliding window. The longest vector remaining is chosen as the dominant orientation. This is then used to orient a square area 20 times the scale the feature point was detected, which is the area covered by the descriptor. This region is spilt up into 4x4 square subregions, from which Haar wavelet responses d_x and d_y are calculated, as well as the sum of absolute values of the responses $|d_x|$ and $|d_y|$ for polarity information. These four responses are calculated for each of the 4×4 subregions, resulting in a 64 value descriptor (4 responses×(4×4) array).

Across the image sequences provided by Mikolajczyk[4], the average number of feature points found by SURF's DoH is similar to those found by SIFT's DoG, as well as the Harris-Laplacian and Hessian-Laplacian detectors (Mikolajczyk and Schmid 2004). SURF was five times faster than the Laplacian detectors, and three times faster than the DoG detector. The SURF descriptor was compared to SIFT, PCA-SIFT and GLOH descriptors, and had the highest recognition rate, with the average SURF recognition rate being 82.6% (85.7% when a slower to compute but more accurate 128 value descriptor was used), compared with 78.3% for GLOH, 78.1% for SIFT, and 72.3% for PCA-SIFT. The repeatability of SURF was comparable or better than the compared algorithms.

While SURF improves on the speed of SIFT while maintaining a comparable level of accuracy, it is still susceptible to perspective deformations, and thus has a limited range of perspective rotation.

2.2.4 Affine Invariant Feature Detectors

The natural feature registration algorithms discussed so far are all susceptible to failure as perspective distortion increases due to movement of the camera

[4] `http://www.robots.ox.ac.uk/~vgg/research/affine`

or marker. All three dimensional world transformations, including changes in perspective, can be modelled by a two dimensional transformation using the projection matrix of the camera. A perspective transformation of a planar surface in three dimensions is modelled by an affine transformation of the same surface in two dimensions. This section examines feature detectors which were designed to be affine invariant and thus perspective invariant.

The previously discussed algorithms fail under perspective distortion due to the affine distortion of the descriptor window. As Baumberg (2000) explains:

"... suppose a circular window centred around a given [feature] point is always used when calculating invariants. After an affine transformation the image structure in the circle is mapped to an elliptical region. If we place a circle around the transformed image feature that contains this elliptical region there will be additional image structures in the region that will distort any invariant measures calculated."

Baumberg (2000) proposes a Harris detector with the "affine Gaussian scale-space" model, described by Lindeberg and Garding (1997), as an affine invariant feature detector. By calculating the shape adapted second moment matrix of a feature point and transforming the feature using the square root of the matrix, a normalized image patch around the feature is obtained. The normalized image patch will be free of affine distortion, and related to the marker feature point by rotation alone. The results of image patch normalization are shown in Figure 2.14. This approach was further developed by Schaffalitzky and Zisserman (2002) for widely separated views of three dimensional geometry.

Mikolajczyk and Schmid (2002) found the features and associated regions used in Baumberg's (2000) approach are "not invariant in the presence of large affine transformations". Their approach was a feature detector capable of estimating any affine transformations affecting the image. A Harris detector is used to locate feature points, and the Laplacian of Gaussian (LoG) is applied for automatic scale selection. Integration and differentiation scales are used to iteratively calculate a shape adaptation matrix of the feature point neighbourhood. Once the iterative process has converged on a shape adaptation matrix, the matrix is used to normalize the feature neighbour-

(a) (b)

Figure 2.14: Normalisation of image patches, (a) Corner features found in two matching image patches, (b) after stretch and skew distortions are removed the patches are only different by rotation (Baumberg 2000)

hood to remove any stretch and skew. The results of this shape adaptation matrix are shown in Figure 2.15. Unfortunately the repeatability of the Harris-Laplace detector is lower than that of SURF, and due to the iterative process of finding a shape adaptation matrix, the performance is significantly lower than any other detector(Bay et al. 2006).

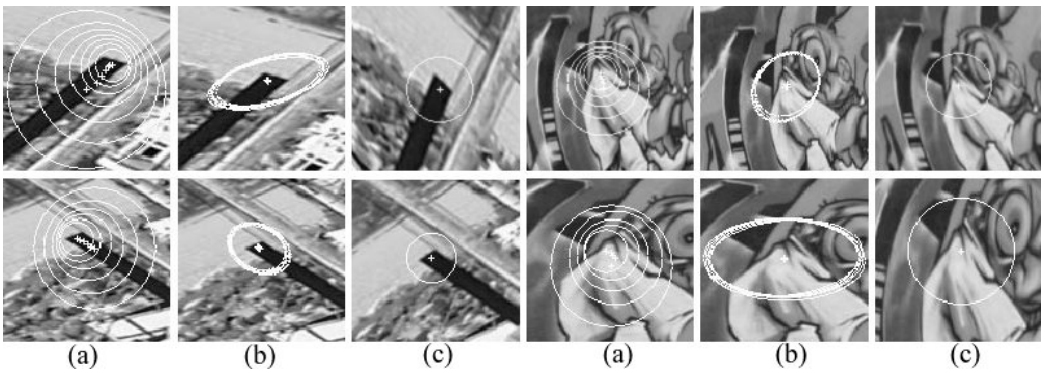


Figure 2.15: An affine invariant feature detector operating on four image features, (a) Multi-scale features found by the Harris detector, (b) The affine regions obtained after multiple iterations, (c) The normalised feature neighbourhoods (Mikolajczyk and Schmid 2002)

The affine distortion of a feature can be calculated by examining the area around a feature point. Brown and Lowe (2002) propose calculation of a feature descriptor based on the region local to multiple feature points which are

nearest neighbours in scale-space, as shown in Figure 2.16. This requires a high level of repeatability of feature detection, but provides an accurate estimation of the two dimensional transformation of a patch. This approach provides a homography on a per feature point basis, allowing erroneous matches to be discarded based on inconsistent transformation estimates, as each homography must align to the object which is being registered. As Brown and Lowe's (2002) method relies on multiple features being correctly detected for calculation of the feature descriptor, the impact of failure to detect a feature is much greater than other natural feature algorithms.

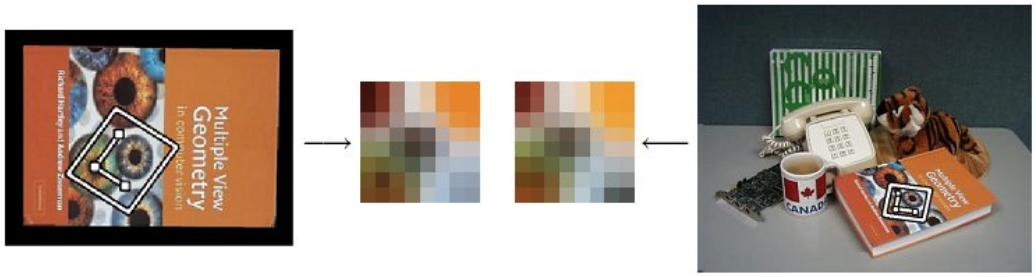
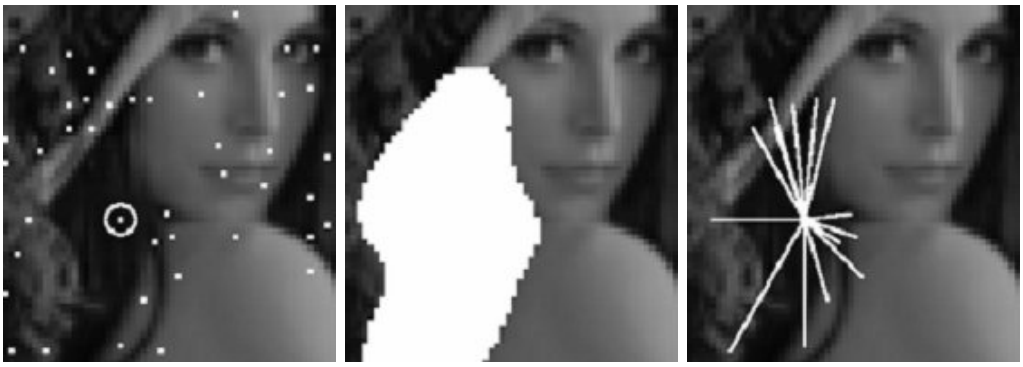


Figure 2.16: Three affine invariant features which are nearest neighbours in scale-space are used to calculate the affine transformation (Brown and Lowe 2002)

The Spider descriptor created by Stanski and Hellwich (2005) is similar to Brown and Lowe's (2002) affine invariant features. Feature points are detected using the SIFT scale space extrema operator, and features with a high reliability score are chosen as the salient points. Around each salient point a region is determined which consists of adjacent pixels with a brightness difference below a defined threshold. From within this region, all feature points which were not reliable enough to be salient are attached to the regions salient point and are termed "anchors". This process is illustrated in Figure 2.17. This method is similar to the one used by Tuytelaars and Van Gool (2000) to find region outlines. When comparing spiders, the weighting of each anchor is determined by its reliability score calculated in the feature detection stage. From there a transformation is estimated which maps one spider to another. The match is successful if the difference between two

spiders is below a threshold.



(a) (b) (c)

Figure 2.17: The determination of a spider, (a) Feature points are selected using the SIFT detector, and a reliable feature is chosen as the salient point for the spider, (b) The region of pixels with a similar average brightness to the salient point are chosen, (c) The feature points within the region in (b) are chosen as the anchor points for the spider (Stanski and Hellwich 2005)

Affine invariant detectors increase the range of registration due to their robustness to perspective deformations of two dimensional objects. Unfortunately full affine invariance has not been accomplished. Despite his own work in the area, Lowe (2004) is critical of affine invariant features, stating "... none of these approaches are yet fully affine invariant ... due to the prohibitive cost of exploring the full affine space. The affine frames are also more sensitive to noise ... so in practice the affine features have lower repeatability than the scale-invariant features unless the affine distortion is greater than about a 40 degree tilt of a planar surface."

Lowe (2004) suggests that a better method of achieving affine invariance is to generate a database of images of the marker after affine transformation, and train each image in the database. This method results in a considerably increased feature database. Despite the criticism of existing affine-invariant descriptors, he believes "affine invariance is a valuable property for matching planar surfaces under very large view changes, and further research should be performed on the best ways to combine this with non-planar 3D viewpoint invariance in an efficient and stable manner."

2.2.5 Feature Classifiers

Lepetit, Pilet and Fua (2004) propose a feature classification approach to natural feature registration which implements Lowe's (2004) suggestion of training affine transformed images of the marker. The marker object is repetitively affine transformed to create a "view-set" of possible transformations, as shown in Figure 2.18. In this view-set, feature points are found by comparing the intensities of pixels on diametrically opposed sides of a circle transcribed around each feature. Features which do not have a grey level which matches that of any surrounding pixels are then thresholded based on the difference between the grey level of the feature and the pixels around the circle. This process removes features in uniform areas and on edges, and selects features with a maximum intensity change.

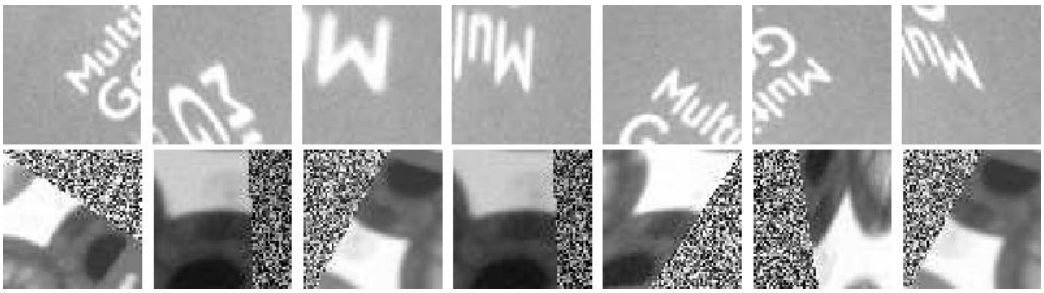


Figure 2.18: Samples of the view sets generated by the feature classifier for two feature points on the cover of a book (Lepetit and Fua 2006)

This process is performed across all images in the view-set. For each image in the view set, the inverse of the transformation used to generate the image from the reference is used to recover the corresponding feature point in the reference image. The total number of images that each feature appears in is counted, and the feature points with the highest number of instances are chosen to be reliable across a range of transformations.

The classification is performed using a series of binary comparisons of intensity over two randomly chosen patches around a feature point. In Lepetit et al.'s (2004) original work, these binary choices are dimensions for Approximate Nearest Neighbours (ANN). A more efficient method was later

proposed using randomized trees (Lepetit and Fua 2006), where each comparison is represented by a branch in the tree. In the training process the surrounding area of each feature point is examined using the set of random patches, and this is used to generate a probability table at each node in the tree. This table determines the probability that a feature point that reaches the node is likely to be that the reference feature point, as shown in Figure 2.19. Several trees are used, each with their own random tables, as the overlap between trees allows for a finer partition and higher accuracy. An interesting outcome of this research was the robustness of the feature matching allowed for the tracking of non-rigid surfaces (Pilet, Lepetit and Fua 2005).

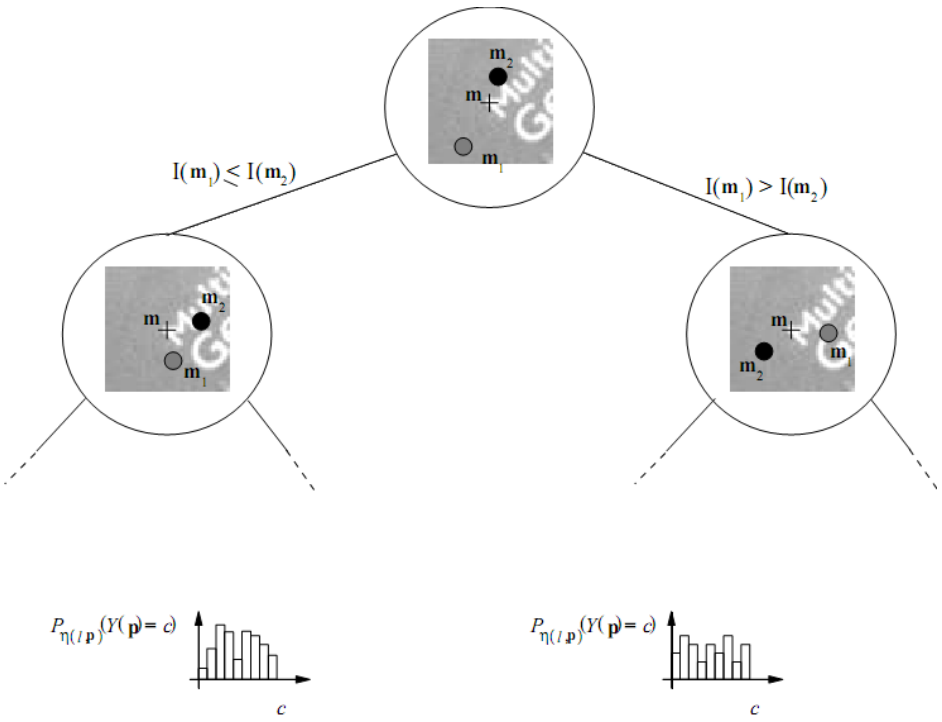


Figure 2.19: The feature classification process. At each level in the tree image patches are observed and the results determine the path taken. At each node there is a probability table of how likely any found feature is to be the reference feature (Lepetit and Fua 2006)

When tested against SIFT, this feature classification approach was found to be more robust to perspective distortion than SIFT, however had fewer

matches when distortion was minimal. The authors admit that SIFT has no training stage, and given a training stage using a similar view-set approach, could prove as robust to perspective distortion as their own method, however maintain that their feature point detection and comparison methods are more computationally effective. Furthermore, the use of a tree structure is not particularly efficient or discriminative when many different classes exist, however this was resolved in a following work with the use of a non-hierarchical data structure known as a Fern (Ozuysal, Calonder, Lepetit and Fua 2009), and a naive Bayesian estimator to calculate the class of a queried feature point (Ozuysal, Fua and Lepetit 2007). In this research we describe this method as the Ferns Classifier. Although this new data structure is able to accurately match features in a large database efficiently, marker training still takes a considerable amount of time.

2.3 Summary

This chapter reviews common planar registration algorithms, including both fiducial and natural feature approaches. Particular attention was paid to the strengths and weaknesses of each approach in order to identify potential areas for improvement, as shown in Tables 2.1-2.2. A common shortcoming in both approaches, but especially prevalent in natural feature registration, is the lack of perspective invariance, causing registration failure when the marker undergoes a large transformation with respect to the camera.

A number of commonalities appear across all planar registration algorithms, the following chapter identifies and provides an in depth examination of aspects which are inherent to planar registration.

Strengths	Weaknesses
Light Beacons (Bajura and Neumann 1995), (Welch et al. 1999), (Ribo et al. 2001), (Mohan et al. 2009)	
Invariant to poor illumination and shadows.	Large size
Visible from far away	Require power source
Infrared invisible markers	May reduce image quality
Fiducial Features (Koller et al. 1997), (Cho et al. 1997), (Cho et al. 1998), (Park et al. 1998)	
Computationally efficient to locate	Robustness depends on lighting
Easy to integrate into environment	Multiple features required for registration
Easy to add scale invariance	Scale invariance relies on features of different size
ARToolKit (Kato and Billinghurst 1999), (Billinghurst et al. 2002)	
Popular and thoroughly reviewed	Highly susceptible to minor occlusion
Single marker gives full 6DOF transformation	Prone to false positive registration
Infinite number of different markers	
ARTag (Fiala 2005a), (Fiala 2005b)	
Highly illumination invariance	Markers have no semantic meaning
Highly occlusion invariant	Limited number of markers
Very few false positive matches	
ARToolKitPlus/Studierstube ES (Wagner and Schmalstieg 2007), (Schmalstieg and Wagner 2007)	
Very computationally efficient	Still prone to fiducial detection failure
Runs on mobile devices	
Provides a range of different marker types	

Table 2.1: Summary of fiducial registration approaches

Strengths	Weaknesses
ARToolKit NFT (Kato et al. 2003), (Taketa et al. 2007)	
Computationally efficient	Requires initialisation from fiducial
Robust to occlusion	Homography calculation from 4 features increases jitter
Scale Invariant Feature Transform (SIFT) (Beis and Lowe 1997), (Lowe 1999), (Lowe 2004), (Ke and Sukthankar 2004), (Grabner et al. 2006), (Lemuz-López and Arias-Estrada 2006)	
Popular and thoroughly reviewed	Computationally expensive
Very accurate registration results	Restrictive Licensing
Scale, rotation and illumination invariant	Not robust to perspective distortion
Speeded Up Robust Features (SURF) (Bay et al. 2006)	
Computationally efficient	Lower accuracy than SIFT
Scale, rotation and illumination invariant	Not robust to perspective distortion
Affine Invariant (Baumberg 2000), (Schaffalitzky and Zisserman 2002), (Brown and Lowe 2002), (Mikolajczyk and Schmid 2002) , (Stanski and Hellwich 2005)	
More robust to perspective distortion	Can be computationally expensive
Wide range of different methods available	Lower accuracy than SIFT
	Not truly perspective invariant
Feature Classifiers (Lepetit et al. 2004), (Pilet et al. 2005), (Lepetit and Fua 2006), (Ozuysal et al. 2007), (Ozuysal et al. 2009)	
Computationally efficient	Long training time
Very good perspective invariance	Large feature database increases matching time/decreases accuracy

Table 2.2: Summary of natural feature registration approaches

Chapter 3

Planar Registration

This chapter examines the theory behind computer vision based planar registration. A mathematical model for the registration process is presented in the context of calculating the transformation between a marker and the camera. The main coordinate systems of the respective transformations are also described. Zhang's (2000) method of camera calibration, which is essential for planar registration, is covered in fine detail, as is the calculation of the registration matrix.

3.1 Context

In this research a camera is defined as a physical device which captures photons on a two dimensional image plane. In the domain of computer vision the camera is comprised of a lens and a digital sensor, which is a grid of receptors which measure the amount of photons. The receptors register the different wavelengths of photons and record a red, green and blue value for each pixel. The number of receptors in the grid determines the digital resolution in pixels of the image, commonly 2^n in each dimension, with the horizontal dimension 4/3 the size of the vertical dimension.

For a theoretical overview of image based registration, a real camera can be approximated mathematically by the pinhole camera model (Xu and Zhang 1996). This model represents a camera with an infinitely small aperture, and is used to define the relationship between a light ray travelling in three dimensions in the world, and the corresponding two dimensional point it forms on the image sensor. Figure 3.1 shows a pinhole camera, with a light ray from a point P passing through the pinhole aperture C (located at $(0, 0, 0)$) and striking the image plane at a point p.

This theoretical model sets the foundation for planar registration. The following section describes the coordinate systems which are used in planar

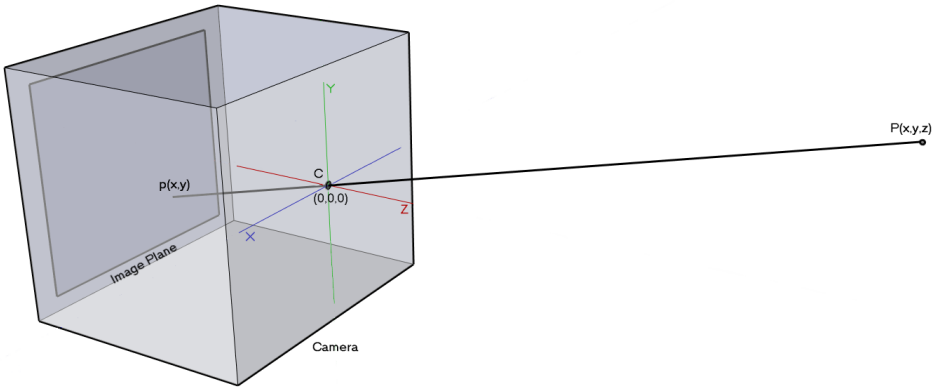


Figure 3.1: The pinhole camera model, showing light travelling from point P through the pinhole aperture at (0,0,0) to strike the image plane at point p. The coordinate system used is the camera coordinate system

registration.

3.2 Coordinate Systems

There are three major coordinate systems in image based registration:

1. The camera coordinate system C_{cs}. This is the three dimensional coordinate system which describes the physical location and orientation of objects with respect to the camera. As shown in Figure 3.1, the origin of the coordinate system is the aperture of the camera, the X and Y axes are parallel to the X and Y axes of the image plane, and the Z axis is the principal axis of the camera. The scale of the coordinate system is in metres.

2. The image coordinate system I_{cs}. This is the two dimensional coordinate system which describes the image projected onto the sensor of the camera, as shown in Figure 3.4. The origin is where the principal axis of the camera intersects the plane (typically the centre), the X axis is the horizontal axis of the image, and commonly called u, and the Y axis is the vertical axis of the image, and commonly called v. The scale of the coordinate system is in pixels.

3. The object coordinate system O_{cs}. This is the three dimensional coordinate system of an object in the world. The origin, axes and scale are dependent on the representation of the object in the system. Registration is the process of finding the transformation which maps the object coordinate system to the camera coordinate system, as shown in Figure 3.3.

Image based registration is the process of calculating the transformations between the three coordinate systems. Figure 3.2 shows the three transformations. Transformation 1 is the object to camera transformation which cannot be measured, but is the desired outcome of registration. This transformation is calculated using the difference between the calculable transformations, transformation 2 and transformation 3. Transformation 2 is the object to image transformation, and transformation 3, the image to camera transformation. The following sections mathematically describe these transformations.

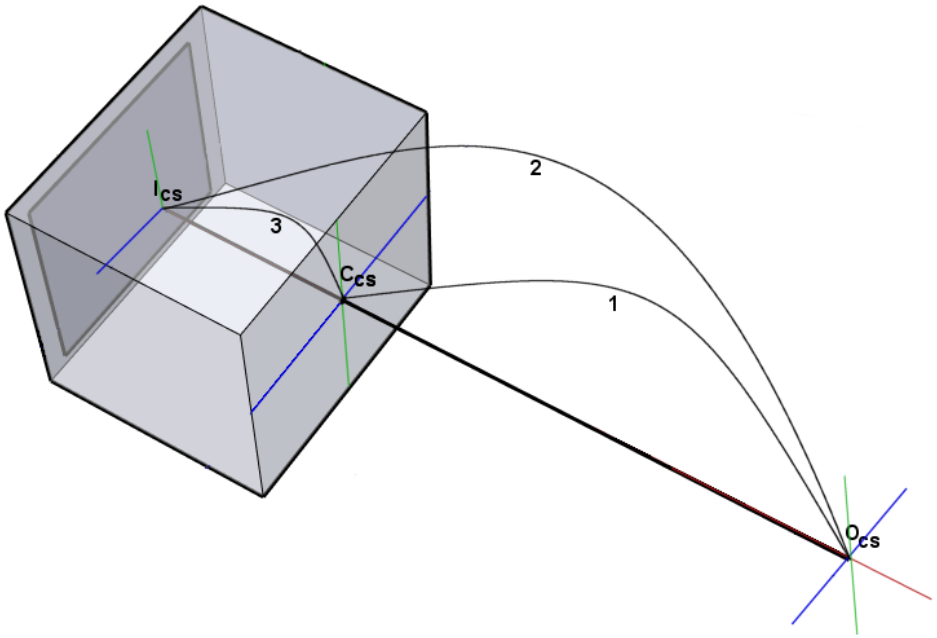


Figure 3.2: The transformations involved in registration: 1 - Object to Camera, 2 - Object to Image, 3 - Image to Camera

3.2.1 Object to Camera Transformation

For an object O which has a position and orientation in three dimensional space, the goal of registration is to find the translation T_x, T_y, T_z, and rotation R_x, R_y, R_z from the object to the camera C. The registration matrix is composed of these six degrees of freedom, three rotational and three translational, which are also known as the "extrinsic camera parameters". Figure 3.3 shows an object coordinate system O_{cs} and the camera coordinate system C_{cs}. In this figure the object is non-planar, to illustrate the orientation of its coordinate system.

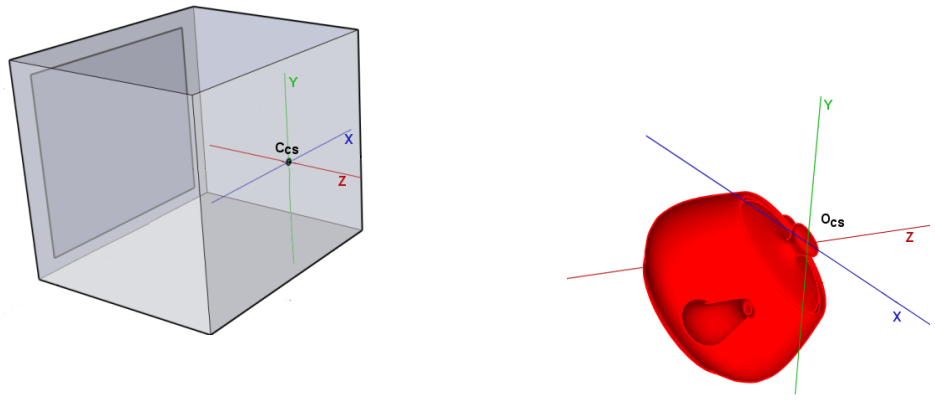


Figure 3.3: Registration is the process of finding the transformation of an object coordinate system, shown here as O_{cs} to a camera coordinate system, C_{cs}

The rigid body transformation from O to C can be represented using a three dimensional translation matrix T and a three dimensional rotation matrix R:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{3.1}$$

$$T = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix} \tag{3.2}$$

The rotational and translational matrices of the extrinsic parameters are usually combined into a 4×4 homography matrix, as shown in Equation 3.3

$$M_{ext} = \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The matrix M_{ext} is the output of the registration algorithm and represents the orientation and position of the object from the perspective of the camera for that moment in time. This transformation cannot be measured, and must be derived from the other two transformations.

3.2.2 Object to Image Transformation

A camera captures images of the three dimensional world on a two dimensional image plane. Photons which have been emitted or reflected from an object pass through the aperture, and strike the image plane, as shown in Figure 3.4.

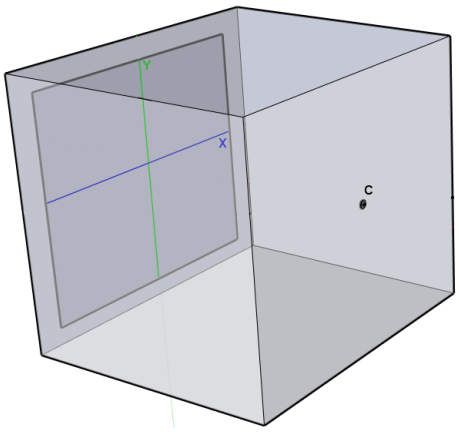


Figure 3.4: The coordinate system of the image plane.

Figure 3.5 illustrates how light from a point P in three dimensional world is represented as a point p in two dimensional image space. For consistency,

points in the three dimensional world are denoted by capital letters, and the corresponding two dimensional pixels on the image plane are denoted by lower case letters. The coordinate system shown is C_{cs}, the Z axis being the principal axis of the camera, and the origin being the camera aperture. In this example a light ray is emitted from or reflected off point P in the world, travels through the aperture, and strikes the image plane at location p. The image plane is at the focal length f of the camera, which is known as "infinite focus".

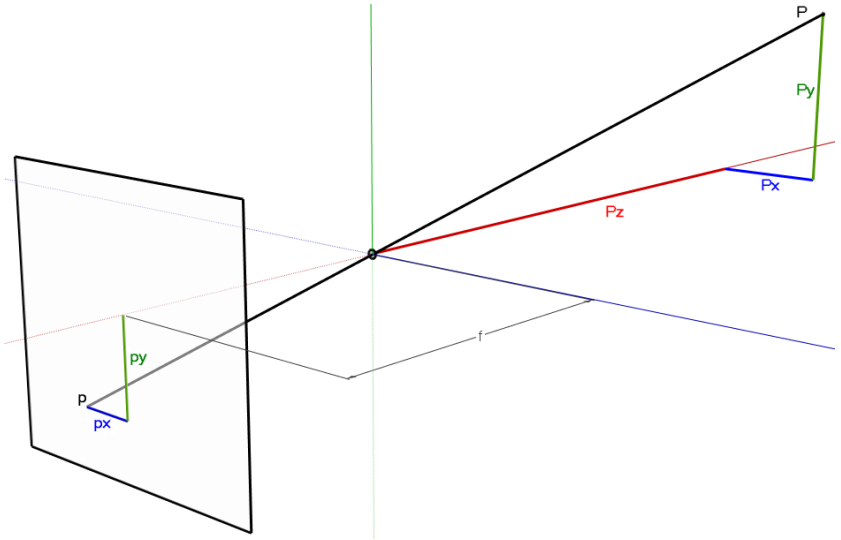


Figure 3.5: The transformation from three dimensional point P to two dimensional point p using the pinhole camera model

Figure 3.6 shows the same example as in Figure 3.5, with the view aligned to the X and Z axes (top) and the X and Y axes (bottom).

The ray which originates from point P and passes through the origin forms two right angle triangles with the C_{cs} axes. The catheti of the triangle between the origin and the image plane are the focal length of the camera f, and the position of intersection of the ray and the image plane p. The catheti of the triangle between the origin and the point P represent the position of point P in C_{cs}. The positions of the point on the image plane are defined as:

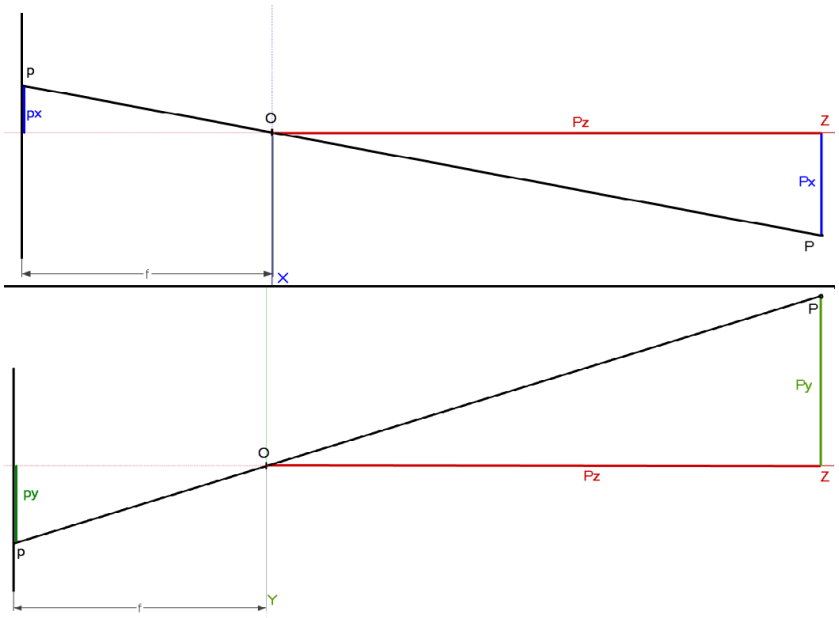


Figure 3.6: The transformation from three dimensional point P to two dimensional point p shown from XZ axis (top) and YZ axis (bottom)

$$\frac{p_x}{f} = \frac{P_x}{P_z} \quad \Rightarrow \quad p_x = f\frac{P_x}{P_z} \tag{3.4}$$

$$\frac{p_y}{f} = \frac{P_y}{P_z} \quad \Rightarrow \quad p_y = f\frac{P_y}{P_z} \tag{3.5}$$

$$p_z = f \tag{3.6}$$

As shown in Figure 3.5, the image which appears on the image plane is flipped horizontally and vertically. This is resolved by flipping the image from the camera such that it has the correct orientation, which assumes a virtual front plane, as shown in Figure 3.7. If the Z coordinate of the point p is represented as $-f$, the virtual front plane has a Z coordinate of f, as shown in Equations 3.4,3.5 and 3.6. All subsequent equations assume a virtual front plane.

Equations 3.4,3.5 and 3.6 can be rewritten in homogeneous matrix form shown in 3.7. This matrix defines the transformation between the three dimensional point P and the two dimensional point p in C_{cs}.

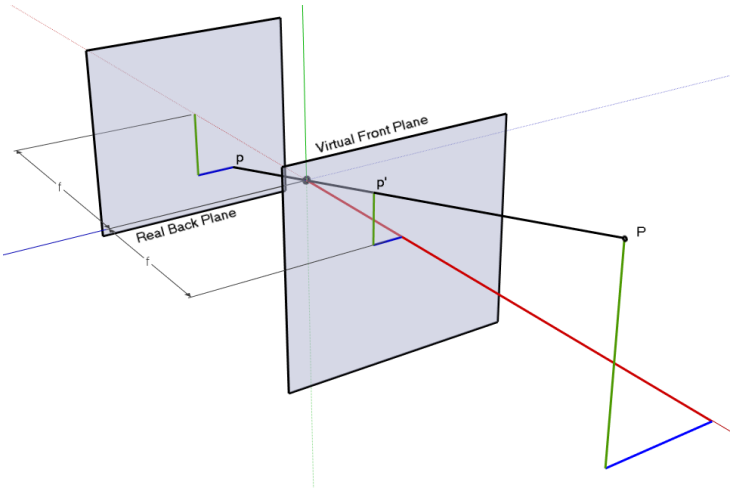


Figure 3.7: The back image plane compared to the virtual front image plane. Point p' is analogous to P, while p is the inverse of P

$$s \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \tag{3.7}$$

The relationship mapping p to P is not linear, as the point on the image plane represents a vector which the point P lies on, rather than a discrete point in three dimensional space. This is represented in Equation 3.7 as the scale factor s.

3.2.3 Image to Camera Transformation

Each camera has intrinsic parameters unique to its physical construction and calibration. The intrinsic parameters are used to find the transformation from C_{cs} to I_{cs}. For planar registration there are six important parameters used in planar registration:

1. Focal length f. As shown in Figure 3.6 and Equations 3.4, 3.5 and 3.6, the focal length f is used in the equations to calculate the position of

the point on the image plane which represents a point in O_{cs}.

2. Principal point u_p, v_p. The principal point represents the point where a ray parallel to the principal axis of the camera passing through the origin strikes the image plane. In the pinhole camera model the principal point is the centre of the image plane, however this is usually not true in practice.

3. Scale factors s_x, s_y. The scale factor which converts from pixel distance to a measurement in the real world at the scale s.

4. Geometric distortion k. While the pinhole camera model assumes no distortion, real cameras focus light through a lens. Depending on the quality of the lens, the resulting image can be distorted, as seen in Figure 3.8. The geometric distortion parameter is used to remove the effects of lens distortion.

The focal length is used to transform a three dimensional point P in the world to a two dimensional point p on the image sensor in C_{cs}. The point p cannot be measured, and must be found by the pixel (u, v) which represents it in the image. Point p is transformed to pixel (u, v) by the principal point (u_p, v_p) and the scale values s_x, s_y using the formulas:

$$p_x = (u - u_p)1/s_x \tag{3.8}$$

$$p_y = (v - v_p)1/s_y \tag{3.9}$$

Equations 3.8 and 3.9 are represented in homogeneous form as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & u_p \\ 0 & s_y & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \tag{3.10}$$

Combining Equations 3.10 and 3.7 together, the transformation from a three dimensional point P in the world to a two dimensional pixel (u, v) in an

image is:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & u_p \\ 0 & s_y & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \tag{3.11}$$

$$\tag{3.12}$$

Which can be further simplified to:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & u_p \\ 0 & s_y f & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \tag{3.13}$$

$$\tag{3.14}$$

The intrinsic camera parameters represent the matrix:

$$M_{int} = \begin{bmatrix} s_x f & 0 & u_p \\ 0 & s_y f & v_p \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

The values of f, (u_p, v_p), s_x, s_y and k are obtained by camera calibration.

3.3 Camera Calibration

There are six internal camera intrinsic parameters which are required to calculate the transformation from the camera to the object: the focal length f, scale factors s_x, s_y, principal point (u_p, v_p), and geometric distortion k. With the assumption that these parameters remain constant, these parameters can be estimated by an off-line camera calibration step. The calibration works by comparing known features on a two dimensional calibration pattern with their locations in the image frame.

There are several calibration methods available; for this research the well known method by Zhang (2000) is used. This method uses an additional parameter, λ, which refers to the skew between two image axes. For simplicity in this research image axes are assumed to be orthogonal and λ is set to zero. The following section describes this calibration process.

3.3.1 Calculation of Intrinsic Parameters

A planar calibration pattern is captured with the camera, with the Z coordinate of each calibration point in C_{cs} assumed to be zero. For each calibration point P on the calibration pattern, the corresponding pixel (u, v) in the image is found using Equation 3.17:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_{int} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ 0 \\ 1 \end{bmatrix} \tag{3.16}$$

$$= M_{int} \begin{bmatrix} r_1 & r_2 & t_1 \\ r_4 & r_5 & t_2 \\ r_7 & r_8 & t_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} \tag{3.17}$$

For every calibration point P, there is a homography H which maps P to its corresponding pixel (u, v), such that:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix} \tag{3.18}$$

Equations 3.17 and 3.18 show that H is equivalent to the camera intrinsic parameters and camera extrinsic parameters combined on the plane at $Z = 0$.

From Equation 3.18, the following can be derived:

$$su = h_{11}P_x + h_{12}P_y + h_{13} \tag{3.19}$$
$$sv = h_{21}P_x + h_{22}P_y + h_{23} \tag{3.20}$$
$$s = h_{31}P_x + h_{32}P_y + h_{33} \tag{3.21}$$

Substituting Equation 3.21 into Equations 3.19 and 3.20 gives:

$$u(h_{31}P_x + h_{32}P_y + h_{33}) = h_{11}P_x + h_{12}P_y + h_{13} \tag{3.22}$$
$$v(h_{31}P_x + h_{32}P_y + h_{33}) = h_{21}P_x + h_{22}P_y + h_{23} \tag{3.23}$$

Equation 3.22 and 3.23 can be rearranged to form:

$$h_{11}P_x + h_{12}P_y + h_{13} - u(h_{31}P_x + h_{32}P_y + h_{33}) = 0 \tag{3.24}$$
$$h_{21}P_x + h_{22}P_y + h_{23} - v(h_{31}P_x + h_{32}P_y + h_{33}) = 0 \tag{3.25}$$

Which are then combined to give:

$$\begin{aligned}(h_{11}P_x + h_{12}P_y + h_{13} - u(h_{31}P_x + h_{32}P_y + h_{33})) + \\ (h_{21}P_x + h_{22}P_y + h_{23} - v(h_{31}P_x + h_{32}P_y + h_{33})) = 0\end{aligned} \tag{3.26}$$

If the homography is represented in vector form as:

$$H_{vec} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} \tag{3.27}$$

then the matrix:

$$A = \begin{bmatrix} P_x & P_y & 1 & 0 & 0 & 0 & -uP_x & -uP_y & -u \\ 0 & 0 & 0 & P_x & P_y & 1 & -vP_x & -vP_y & -v \end{bmatrix} \tag{3.28}$$

Multiplies with H_{vec} to form Equation 3.26, making Equation 3.29 true:

$$AH_{vec} = 0 \tag{3.29}$$

The value of H_{vec} calculated from Equation 3.29 is only true for a given point P and its corresponding pixel (u, v). With at least four pairs the equation:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} H_{vec} = 0 \tag{3.30}$$

can be solved using singular value decomposition for total least squares minimization. The solution is the right singular vector of A corresponding to the smallest singular value (Zhang 2000). As the result of this may not be perfect due to corruption by noise in the image points, the maximum likelihood estimation can be found using the Levenberg-Marquardt algorithm (Moré 1978) on an approximation of the noise.

The calculation of the intrinsic camera parameters requires decomposition of multiple homographies. If the homography is represented as the matrix of vectors $H = [h_1 h_2 h_3]$, Equation 3.17 can be written as

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = M_{int} \begin{bmatrix} r_1 & r_2 & t_1 \\ r_4 & r_5 & t_2 \\ r_7 & r_8 & t_3 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.31}$$

Given that $r_{1,4,7}$ and $r_{2,5,8}$ are orthonormal, the following is true:

$$h_1^T (M_{int}^{-1})^T M_{int}^{-1} h_2 = 0 \tag{3.32}$$

$$h_1^T (M_{int}^{-1})^T M_{int}^{-1} h_1 = h_2^T (M_{int}^{-1})^T M_{int}^{-1} h_2 \tag{3.33}$$

A new matrix B is defined such that:

$$B = (M_{int}^{-1})^T M_{int}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \tag{3.34}$$

From calculating the matrix $(M_{int}^{-1})^T M_{int}^{-1}$ it is shown that B is symmetrical along the main diagonal, and can be represented by a 6 dimensional vector $b = [B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33}]$.

If the i^{th} column vector of H is $h_i = [h_{i1}, h_{i2}, h_{i3}]$ then

$$h_i^T B h_j = v_{ij}^T b \tag{3.35}$$

Where v_{ij} is found by

$$v_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix} \tag{3.36}$$

As $B = (M_{int}^{-1})^T M_{int}^{-1}$, Equation 3.32 can be rewritten as:

$$h_1^T B h_2 = 0 \tag{3.37}$$

and Equation 3.33 can be rewritten and reordered as:

$$\begin{aligned} h_1^T B h_1 &= h_2^T B h_2 \\ h_1^T B h_1 - h_2^T B h_2 &= 0 \end{aligned} \tag{3.38}$$

Equations 3.37 and 3.38 can be combined with Equation 3.35 to form

$$v_{12}^T b = h_1^T B h_2 = 0 \tag{3.39}$$

$$(v_{11} - v_{22})^T b = h_1^T B h_1 - h_2^T B h_2 = 0 \tag{3.40}$$

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \tag{3.41}$$

As multiple points were used to calculate a single homography in Equation 3.30, so can multiple homographies from multiple images of the calibration pattern be used to calculate b. With a minimum of 2 images, assuming that skew λ is 0 a unique solution can be found for b; the solution is the right singular vector of $(v_{12}, v_{11} - v_{22})$ corresponding to the smallest singular value.

The matrix B can be generated from the now known values of b. The values for M_{int} are found as follows:

$$v_p = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \tag{3.42}$$

$$s = B_{33} - [B_{13}^2 + v_p(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \tag{3.43}$$

$$s_x = \sqrt{s/B_{11}} \tag{3.44}$$

$$s_y = \sqrt{sB_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{3.45}$$

$$u_p = -B_{13}s_x^2/s \tag{3.46}$$

The solution for M_{int} can be refined using a non-linear minimization algorithm, such as the Levenberg-Marquardt algorithm (Moré 1978), on Equation 3.47:

$$\sum_{i=1}^{N}\sum_{j=1}^{M} ||p_{ij} - \hat{m}(M_{int}, R_i, t_i, P_j)||^2 \tag{3.47}$$

where N is the number of images captured of the calibration pattern, and M is the number of calibration points. The extrinsic parameter R_i is a Rodrigues rotation vector parallel to the rotation axis and of magnitude equal

to the angle, and the parameter t is the translation vector. The formulae for calculating these extrinsic parameters are shown in Equations 3.53-3.55. The refinement takes three to five iterations to converge, and provides a very consistent value for M_{int} (Zhang 2000).

Camera calibration requires non-linear minimisation, which is a computationally intensive process. By performing an off-line calibration of the internal parameters the computation required for registration is vastly reduced. Assuming all the camera intrinsic parameters remain constant, the calibration only needs to be performed once for a camera.

3.3.2 Geometric Distortion Removal

Although the pinhole camera model has no image distortion, the lens used in real cameras can introduce radial and tangential distortion. Although tangential distortion can be ignored for most vision algorithms (Tsai 1987), radial distortion produces noticeable curvature effects in the image such as barrel distortion. For example, Figure 3.8 is an image taken of a grid of straight lines with an uncalibrated ADS USB2.0 Turbo WebCam, as described in Section 7.1.1. The image shows significant radial barrel distortion.

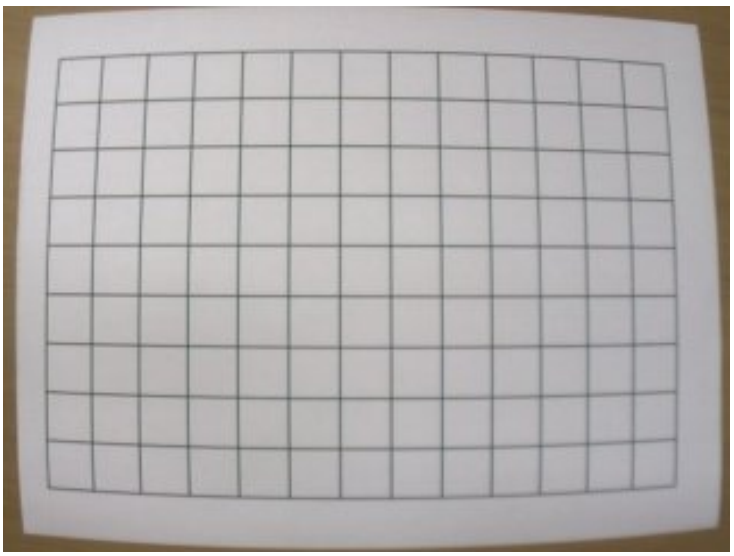


Figure 3.8: Barrel distortion effect in an uncalibrated digital camera.

Equations 3.48, 3.49 describe the transformation of a pixel X_u, Y_u position by radial distortion.

$$X_d = X_u - X_d D \tag{3.48}$$
$$X_y = Y_u - X_y D \tag{3.49}$$

Where X_d, Y_d is the measured distorted pixel location, and D is the infinite series:

$$D = k_1 r^2 + k_2 r^4 + ... + k_\infty r^\infty \tag{3.50}$$
$$r = \sqrt{X_d^2 + Y_d^2}$$

The values of k_n represent the distortion coefficients of the lens. Tsai (1987) states that in practice only a single term $D(n = 1)$ is needed unless using a wide angle lens, and any more elaborate modelling would cause numerical instability.

The value of k_1 is found by introducing it into Equation 3.47 and minimizing again, using the estimates obtained previously as the values of M_{int}, R_i and t_i, as shown in Equation 3.51.

$$\sum_{i=1}^{N} \sum_{j=1}^{M} ||p_{ij} - \hat{m}(M_{int}, k_1, R_i, t_i, P_j)||^2 \tag{3.51}$$

Once the distortion coefficient is found, the inverse of Equations 3.48, 3.49 and 3.50 are applied to an image to remove any radial distortion effects, providing a reliable assumption that the captured image is free from radial distortion, and fits the pinhole camera model.

3.4 Registration Calculation

The outcome of planar registration is the transformation from the object to the camera. As previously described, this transformation is found using the combination of the transformations of the object to the image, and the image to the camera. The previous section described the process of calculating the

image to camera transformation to obtain the intrinsic camera parameters. Once the transformation of the points from the object to the image is known, the extrinsic camera parameters can be calculated.

The rotation and translation parameters which make up the object to camera transformation are calculated using Equations 3.53, 3.54, 3.55 and 3.55, using the homography of the object to the image plane calculated in Equation 3.30, and the camera's intrinsic parameters.

$$\begin{aligned} \lambda &= 1/||M_{int}^{-1}h_1|| \\ r_1 &= \lambda M_{int}^{-1}h_1 && (3.52) \\ r_2 &= \lambda M_{int}^{-1}h_2 && (3.53) \\ r_3 &= r1 \times r2 && (3.54) \\ t &= \lambda M_{int}^{-1}h_3 && (3.55) \end{aligned}$$

Due to the presence of noise, the orthogonality of R may not be true. In order to enforce orthogonality, singular value decomposition is used to find the orthogonal matrix R' which has the smallest Frobenius norm difference, which is the square root of the sum of absolute squares of each element, to R. R' must fulfil the following properties:

$$\begin{aligned} R' &= min||R' - R||_F^2 && (3.56) \\ ||R' - R||_F &= \sum_{i=1}^{3}\sum_{j=1}^{3} R'_{ij} - R_{ij} && (3.57) \\ R'^T R' &= I && (3.58) \end{aligned}$$

3.5 Summary

In this chapter, the coordinate systems important for registration were presented. Use of the Zhang (2000) method of camera calibration to obtain the intrinsic camera parameters and geometric distortion coefficients was explained, as was how to use these values to remove geometric distortion and calculate the registration matrix. Methods for ensuring orthogonality and

minimising error in the resulting calculations were also discussed.

As described in Section 3.4, to find the registration matrix the homography which transforms the planar object to the image must be known. The homography calculation, shown in Equations 3.27-3.30 requires matching points between the object and image. The methods used by natural feature registration to find these matching points are discussed in detail in the following chapter.

Chapter 4

Natural Feature Registration

In this chapter natural feature registration is decomposed into processes which result in a matching feature set for registration computation, as explained in Chapter 3. It is the implementation of each process which differentiates between registration algorithms.

In order to calculate the registration matrix, a minimum of four matching features between the marker image and the frame being registered must be known so as to compute the homography. In an off-line training phase, unique features in the marker are identified and evaluated to ensure robustness to common transformations and deformations. While the application is running, unique features in each frame of video are identified, and each feature found is matched against the features found in the marker to find the best match. The matches with a high correlation are used in the homography calculation, while matches with low correlation are discarded. This process is illustrated in Figure 4.1.

Regardless of the specific natural feature registration algorithm used, there are three stages required for natural feature registration:

1. Feature detection. This stage involves identifying unique features which are robust to transformation and deformation within an image.

2. Feature description. Descriptors are generated which identify and differentiate the features found in the detection stage.

3. Feature matching. Once features and their descriptors have been obtained for a marker and frame, feature matching identifies corresponding features in the images. These matches are used for homography estimation and extrinsic camera parameter calculation.

Each of these stages is described in greater detail in the following sections.

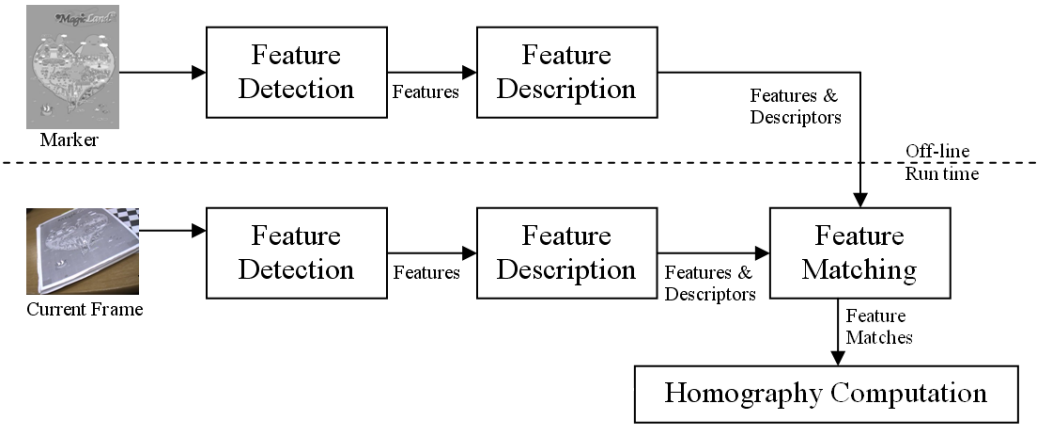


Figure 4.1: The natural feature registration pipeline. Feature detection and description is performed on a marker in an off-line process, and during run time for the current frame. The features are matched, and used for homography computation

4.1 Feature Detection

The first stage of natural feature registration is to locate reliable feature points. A feature point is a structure within an image which has a high uniqueness in the area surrounding it, so that it can be reliably located in the image irrespective of transformation and deformation. Ideal feature points are points in the image which are easily identifiable due to a significant difference with neighbourhood pixels, or the intersection of two lines.

Figure 4.2 shows common transformations of an image, with red squares indicating reliable feature points in each frame. For a feature detector to be robust, it should be invariant to as many transformations as possible. The process of finding feature points is well researched, with many popular detectors such as those by Canny (1986), Harris and Stephens (1988), and Rosten, Porter and Drummond (2008).

Discrete points are implicitly robust to translation, rotation and perspective distortion. However, feature points are not usually scale invariant as shown in Figure 4.2(e). A change in scale may increase the size of the point till it exceeds the feature detector window size, or reduce the size of

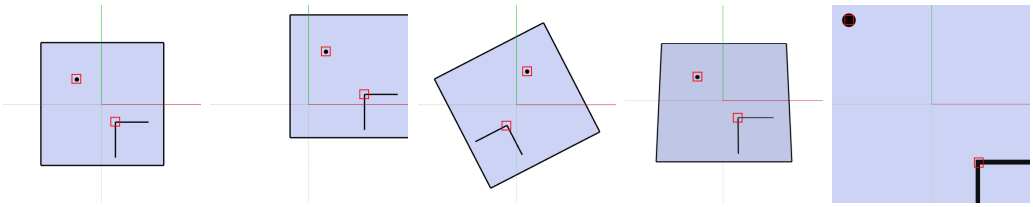


Figure 4.2: Common transformations which feature descriptors can be invariant to, (a) Original image, (b) Translation, (c) Rotation, (d) Perspective distortion, (e) Scaling. Most feature detectors are implicitly invariant to the first three transformations, while a change in scale requires a larger search window.

the point to sub pixel. Invariance to changes in scale is added by performing feature search across multiple scale spaces, using methods such as LoG (Lindeberg 1994), DoG (Lowe 2004), or DoH (Bay et al. 2006). The scale at which the point was detected is also important in the calculation of the descriptor, as in described in Section 4.2.

Figure 4.3 shows the outcome of three popular feature detectors: the SIFT detector using Difference of Gaussian, the SURF detector using Determinant of Hessian, and the FAST corner detector(Rosten and Drummond 2005). The location and count of the feature points vary between algorithms, and the same algorithm will even yield different results depending on its parameters. The SIFT and SURF feature detection algorithms shown in Figure 4.3 discard overlapping features, while the FAST corner detector does not.

Once the registration algorithm has compiled a list of robust feature points, a descriptor is calculated for each feature in order to identify and differentiate them.

4.2 Feature Description

A feature point is represented by a two-dimensional position in an image and, in the case of scale invariant detectors, the scale factor at which the point was identified. Additional information is required in order to differentiate between feature points. In natural feature registration, this information is

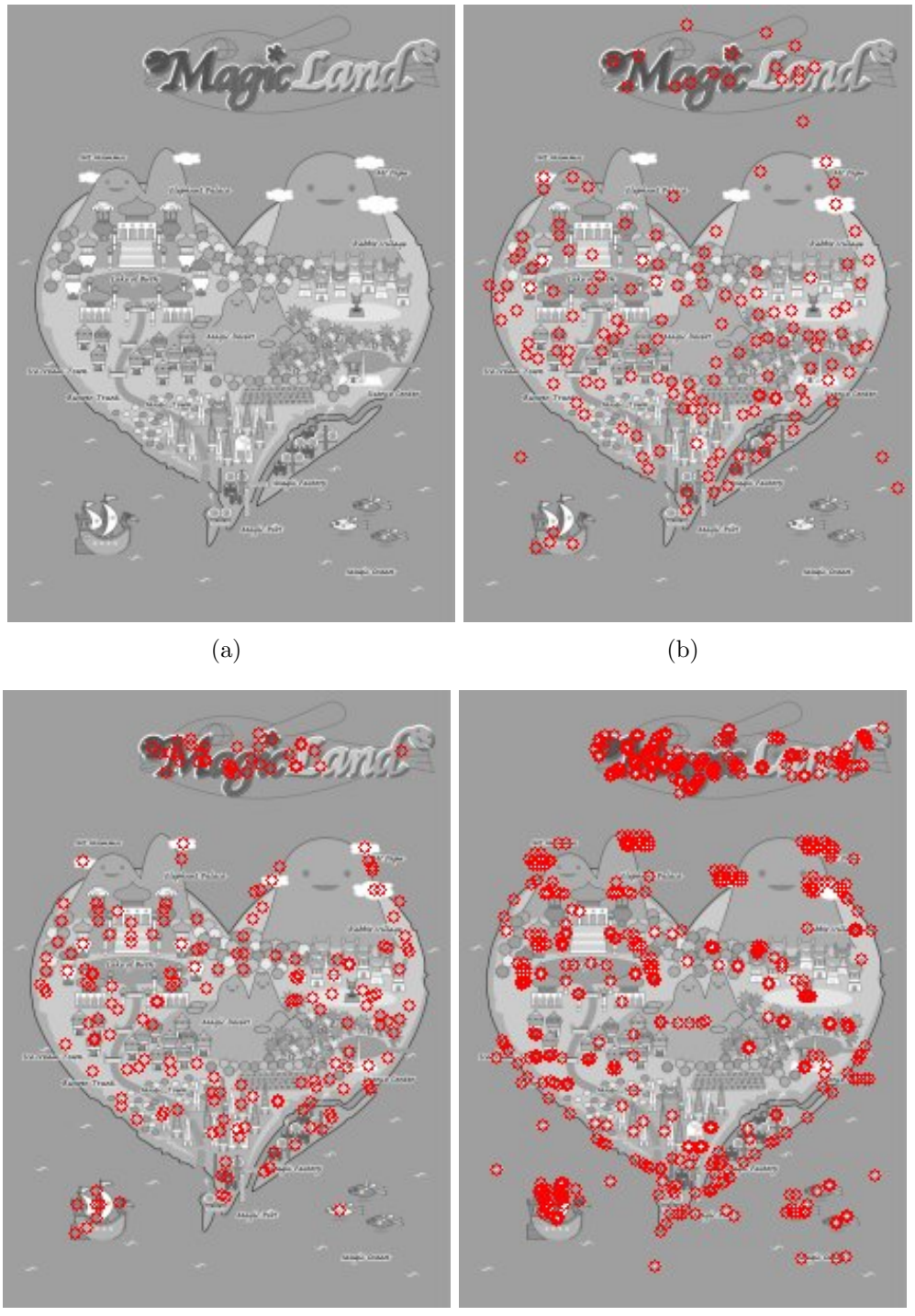


Figure 4.3: A comparison of feature detectors. (a) Original image, (b) the SIFT feature detector, (c) the SURF feature detector, (d) the FAST corner detector. Each feature detector identifies different features. SIFT and SURF do not allow overlap, while the FAST corner detector does

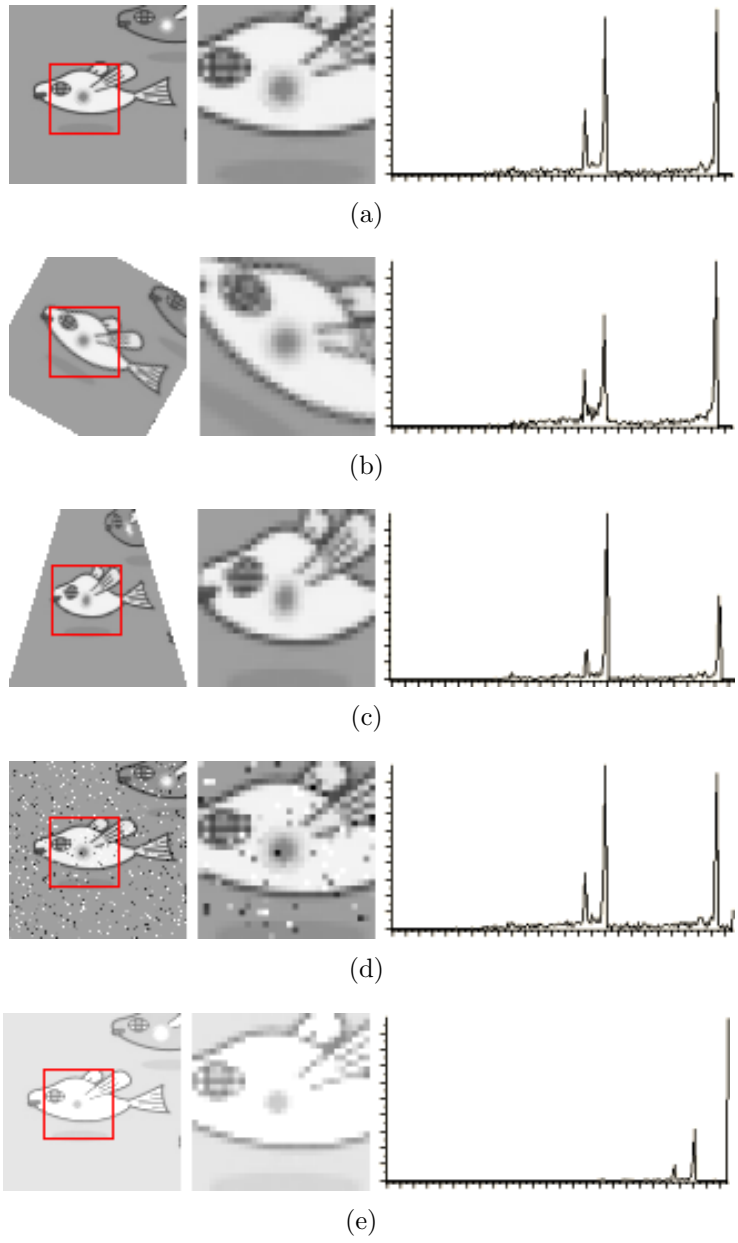


Figure 4.4: Descriptor windows and histograms of a single feature after distortion and transformation. (a) Original image, (b) After 30° rotation, (c) After 60% vertical perspective distortion, (d) After 10% random noise, (e) After 28% brightness increase,

unique to each feature point and is called the feature descriptor. The robustness of the registration algorithm is directly correlated to the quality of the descriptor, and calculation of the feature descriptors is usually the most computationally intensive stage in registration.

To calculate a feature descriptor, a window of the image around a feature point is converted into a data structure using a specialised algorithm which is robust to transformation, distortion, noise and illumination. The algorithms are specific to the registration algorithm, and often there is a compromise between the robustness and differentiability of the feature descriptor.

Using a histogram of illumination of an image as an example of a feature descriptor, Figure 4.4 shows the effect of four common image transformations and rigid body transformations: rotation, perspective distortion, noise and illumination. The illumination histogram is robust to salt and pepper noise and to a lesser extent rotation, but performs poorly in the event of the changes in illumination and perspective.

To ensure robust registration, feature descriptors must be invariant to rigid body transformations. All descriptors are invariant to translation, as the feature descriptor is always centred on the feature point. Scale invariance is inherited from the feature detector, and the window size used for the feature descriptor is determined by the scale space at which the feature was detected. Rotation invariance can be achieved either by using a circular window with a rotationally invariant algorithm, or assigning an orientation for each feature which the window is aligned to, as shown in Figure 4.5.

The level of invariance of the feature descriptor to image distortions such perspective distortion, noise and poor illumination depends on the registration algorithm. Perspective distortion can be minimised using affine transforms, as discussed in Section 2.2.4. The effect of local noise can be reduced by smoothing or averaging the data inside the image window before descriptor calculation. Histogram equalisation of the image data reduces the effect of poor illumination.

The deformation and transformations listed above account for the most common two dimensional transformations and deformations encountered in planar natural feature registration applications.

Figure 4.5: Scale and rotation invariance in the SURF feature descriptor. The size of the window is determined by the feature scale. Each feature has an "upright" orientation, shown by the line (Bay et al. 2006)

4.3 Feature Matching

Once feature points have been identified and descriptors calculated for the marker and a video frame, the final stage of natural feature registration is to find feature points which exist in both images during the feature matching stage. Figure 4.6 shows how a set of features from the marker and a video frame are matched.

The simplest method of feature matching is a linear search through both sets of feature points, using a sum of squares comparison of each element in the descriptors. Each feature pf in the set of all frame features P_f is compared against each feature pm in the set of all marker features P_m. The sum of squares of the difference between all n elements of the descriptors *.desc* of pf and pm is calculated, and if the sum of differences falls below a threshold ϵ, the two features are considered a match. Algorithm 4.3.1 shows this linear sum of squares search.

The sum of squares of the difference between the descriptors is referred to as the "score" of the match, with a lower score indicating a better similarity between two points. For each feature point pf which exists in P_F, Algorithm

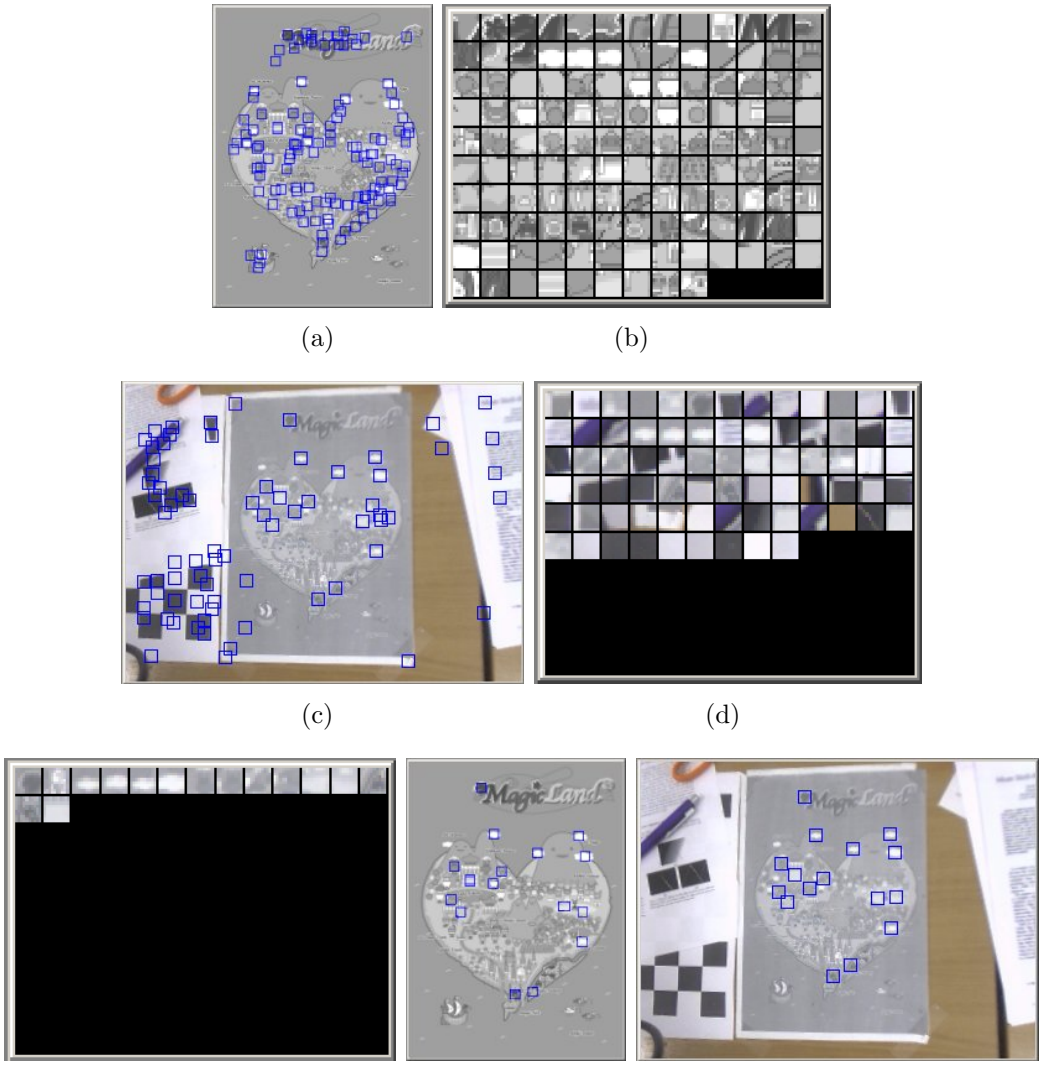


Figure 4.6: Results from a feature matching algorithm (a) Features from Original Image, (b) Feature list from original image, (c) Features from frame, (d) Feature list from frame, (e) Features positively matched from (b) and (d), (f) Matched features from (a), (g) Matched features from (c)

Algorithm 4.3.1: LINEAR SUM OF SQUARES SEARCH(P_m, P_f)

for each pf **in** P_f
  $pf.bestMatch =$ **null** ;
  $pf.bestScore = \epsilon$;
  for each pm **in** P_m
    $score = \sum_{i=1}^{n}(pm.desc[i] - pf.desc[i])^2$;
    if $score < pf.bestScore$ **then**
      $pf.bestMatch = pm$;
      $pf.bestScore = score$;
  next
next

4.3.1 finds the matching feature point pm which exists in P_M with the best score. If feature point pf is very similar to two or more feature points in P_M, the matching feature point found may not be optimal due to noise or distortion. Even a single false match can lead to a serious miscalculation of the homography and large errors in registration.

To reduce the occurrence of false matches, a better linear sum of square search is described in Algorithm 4.3.2. The best and second best matches for each feature point pf are found in the same manner as Algorithm 4.3.1. An additional step compares the ratio of the best match to the second best match, if the ratio is below a threshold ϵ_2 the match is discarded. This ensures that all matches are optimal, with little chance of a false positive match.

Algorithms 4.3.1 and 4.3.2 operate in O($n \times m$) time, where n is the number of feature points in the marker and m the number of feature points in the current frame. The complexity of the algorithms rise exponentially as the number of features in the marker and current frame increase, making the algorithms unsuitable for real-time applications unless the number of features found are low, which causes a decrease in robustness of registration.

A more efficient method of feature matching is a nearest neighbour search using kd-trees (Bentley 1975), which iteratively divides n-dimensional space such that all nodes in the left half of the division are less than the value at

Algorithm 4.3.2: BETTER LINEAR SUM OF SQUARES SEARCH(P_m, P_f)

```
for each pf in P_f
   pf.bestMatch = null ;
   pf.bestScore = ε;
   pf.secondBestMatch = null ;
   pf.secondBestScore = ε;
   for each pm in P_m
      score = Σ_{i=1}^{n} (pm.desc[i] − pf.desc[i])^2;
      if score < pf.bestScore then
         pf.secondBestMatch = pf.bestMatch;
         pf.secondBestScore = pf.bestScore;
         pf.bestMatch = pm;
         pf.bestScore = score;
       else if score < pf.secondBestMatch then
         pf.secondBestMatch = pm;
         pf.secondBestScore = score;
   next
next

for each pf in P_f
   if pf.bestMatch/pf.secondBestMatch < ε_2 then
      pf.bestMatch = null ;
next
```

the division and vice versa. Optimally nearest neighbour searches operate at O(log$n$) when searching, however this speed decreases with an increase in the dimensionality of the descriptor, described by Bellman (1961) as the "curse of dimensionality" . This can be minimised by using approximate nearest neighbour algorithms (Indyk and Motwani 1998), which reduce the search space at the cost of accuracy of the final result, or Best Bin First (Beis and Lowe 1997), a variant of kd-trees designed for indexing higher dimensional spaces, but with slower matching times.

4.3.1 RANSAC

Once feature matching has concluded, a complete set of matched feature points from the marker and the current frame are known. Each match represents two features with the minimum difference in the descriptors, however false positive matches may be present, which can have serious impact on the accuracy of the registration. These outlier matches can be removed using Random Sample Consensus (RANSAC) (Fischler and Bolles 1981).

RANSAC works by randomly choosing a set of data points, constructing a line of best fit for the data points, and testing the remaining points against the model. If a sufficient number of points fit the model it is considered accurate and if not another set of points are chosen and the process is repeated. If the model is accurate, all the points which fit the model, known as inliers, are used to refine the model. The refined model is tested and evaluated based on the error of the inliers to the model. The algorithm concludes after the error drops below a certain threshold, or after a defined number of iterations, in which case the best estimate found is used.

RANSAC can be extended to calculate homographies based on a minimum of four point matches. Four matched feature pairs are chosen randomly, and the homography H is calculated, using the formulas shown Equations 3.27 to 3.30. The other matched feature pairs are then evaluated by calculating the point p on the marker that point P in the frame maps to. If the difference between these two points falls below a threshold ϵ the point is considered good, as shown in Equation 4.1.

$$p - HP < \epsilon \tag{4.1}$$

Once the inliers calculated in Equation 4.1 have been found, the homography is refined using all inliers with Equation 3.30. The model is evaluated by the error of all inliers compared to the models prediction, as shown in Equation 4.2.

$$\sum (p - HP) < \epsilon \tag{4.2}$$

If the model is accurate enough, the homography computed can then be

used to calculate the extrinsic camera parameters using Equations 3.53, 3.54, 3.55 and 3.55.

Although RANSAC is an iterative algorithm, if the ratio of accurate matches to erroneous matches is high, convergence is typically reached after a few iterations. It is an attractive solution for outlier removal in registration as it calculates the optimal homography which can be used for the computation of the extrinsic camera parameters.

4.4 Summary

In this chapter, natural feature registration was decomposed into its fundamental stages, and each stage was thoroughly described. The final result of natural feature registration is a set of feature matches which can be used to calculate a homography and thus the extrinsic camera parameters (as discussed in Chapter 3).

In the feature detection and description stages, several common image transformations and distortions were identified which can reduce the accuracy of registration. In the following chapters, the effects of these transformations and distortions are precisely examined, and solutions to minimise the effects are proposed.

Chapter 5

OPIRA: The Optical-flow Perspective Invariant Registration Augmentation

In the previous chapter, natural feature registration was decomposed into its three main stages; feature detection, description and matching, and the weaknesses of each stage to image transformations and distortions were examined. In the following two chapters, methods to remove the effects of these transformations and distortions are presented.

The first two stages of registration, feature detection (Section 4.1) and feature description (Section 4.2), are responsible for the majority of the robustness of a natural feature registration algorithm. The success of feature detection and description depends in part on the invariance of the algorithms to common image transformations and distortions, identified in the previous chapter as:

1. Rotation. As feature points are zero dimensional, feature detection is rotation invariant. Feature descriptors may not be rotation invariant depending on the registration algorithm.

2. Scale. Scale invariance is important for feature detection. Invariance to changes in scale for feature description is usually handled by calculating the descriptor at the scale space the feature was detected at.

3. Perspective. As the camera rotates away from perpendicular with the marker, perspective distortion increases. Feature detectors are not affected, but the descriptors calculated have less correlation with those found in the marker image.

4. Noise. In this research noise is defined as corruption during the capture and conversion of optical information to digital information. Under this

definition, noise includes any corruption caused by the hardware of the camera such as out of focus blur, and inaccuracies during conversion to a digital signal, such as salt and pepper noise. Noise affects feature detection and description.

5. Illumination. Many of the registration algorithms investigated use illumination information in the identification of features and the calculation of descriptors.

The final stage of natural feature registration, known as feature matching (Section 4.3), relies on a high correlation between the features from the marker and the frame for optimal registration accuracy. When the marker and frame are captured using two different sources, the correlation may not be optimal.

In this chapter a new method of natural feature registration called the Optical-flow Perspective Invariant Registration Augmentation (OPIRA) (Clark, Green and Grant 2008) is presented. OPIRA improves rotation, scale and perspective invariance on prior research. The effects of Noise, Illumination, and Marker sources are investigated in Chapter 6.

Natural feature registration algorithms are most accurate when the view of the marker from the camera's perspective is similar to the image of the marker that is to be registered. As the camera rotates away from being perpendicular with the marker, the effect of perspective distortion on the marker increases which reduces correlation between features in the frame and the marker. The reduced correlation is due to the effect perspective distortion has on the descriptor window used in the feature descriptor stage, as shown in Figure 5.1, and described in further detail in Section 4.2.

The effect of perspective distortion on the descriptor window was explained by Baumberg (2000): "... suppose a circular window centred around a given [feature] point is always used when calculating invariants. After an affine transformation the image structure in the circle is mapped to an elliptical region. If we place a circle around the transformed image feature that contains this elliptical region there will be additional image structures in the region that will distort any invariant measures calculated."

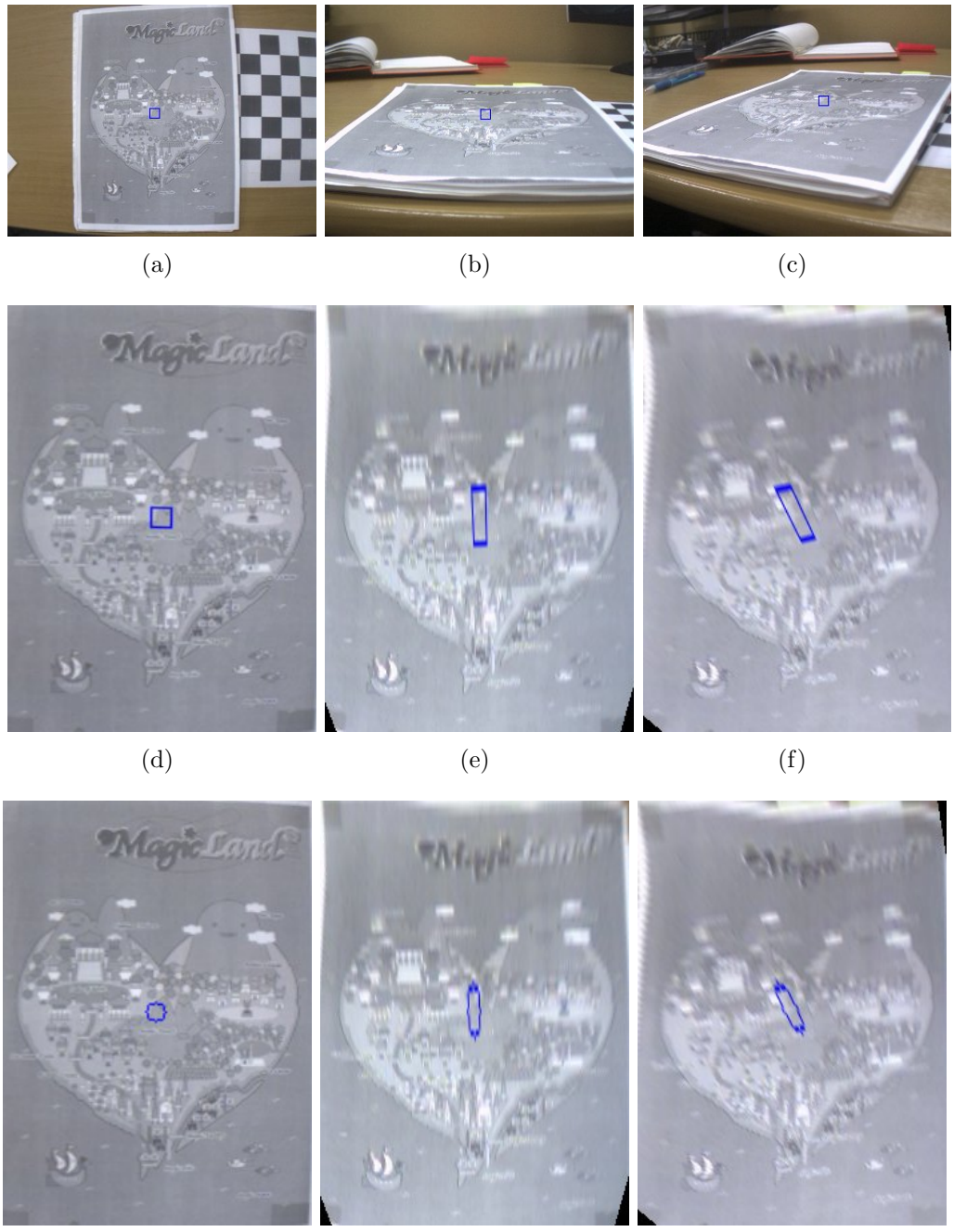


Figure 5.1: The effect of perspective distortion on descriptor windows, (a-c) Transformations of the camera around the marker, with window around feature point relative to image plane, (d-f) The rectified marker, showing the actual shape of the descriptor window relative to the marker, (g-i) The same as above, but showing the effect on a circular window

The OPIRA method overcomes the negative effect of perspective distortion by rectifying the marker such that it is always parallel to the camera's image plane. This rectification additionally adds scale and rotation invariance if not already present in the registration algorithm. The following sections revise the method of standard natural feature registration, and introduce the optical flow method, a common addition to natural feature registration applications. Finally, OPIRA is described in the context of how it functions in unison with these methods to add scale, rotation and perspective invariance to natural feature registration.

5.1 Standard method of Natural Feature Registration

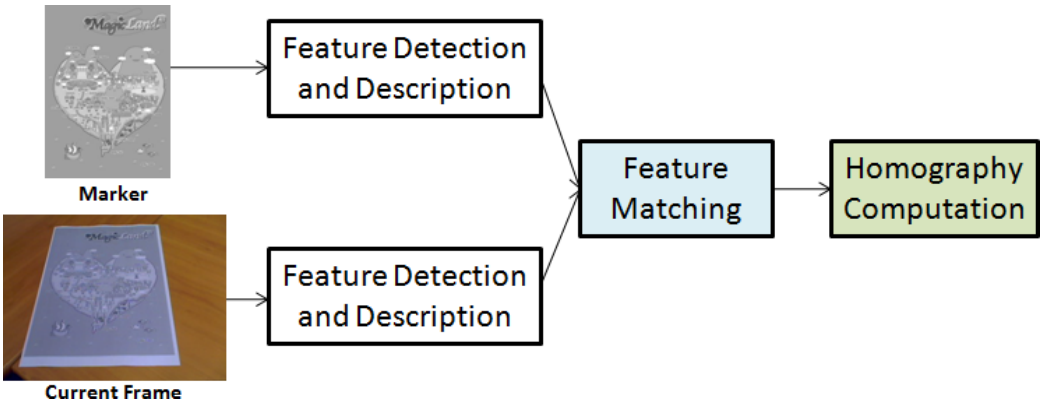


Figure 5.2: The standard method of natural feature registration. Feature detection and description is performed on the marker image and current frame. These features are then matched, and the matches are used to compute a homography.

The standard method of natural feature registration consists of nothing more than the three stages of NFR and homography computation, as discussed in detail in Chapter 4. As illustrated in Figure 5.2, feature detection and description is performed on both the marker, as described in Sections 4.1 and 4.2, and the current frame of video. The features extracted from these steps are then matched together, and these are used to compute a homography which describes the transformation of the frame to the markers. This

homography can then be used to find the transformation using the camera intrinsic parameters, and described in Section 3.4.

The feature matching and homography computation stages are highlighted to show how they are connected to the optical flow and OPIRA methods of registration in the following sections.

5.2 Optical Flow method of Natural Feature Registration

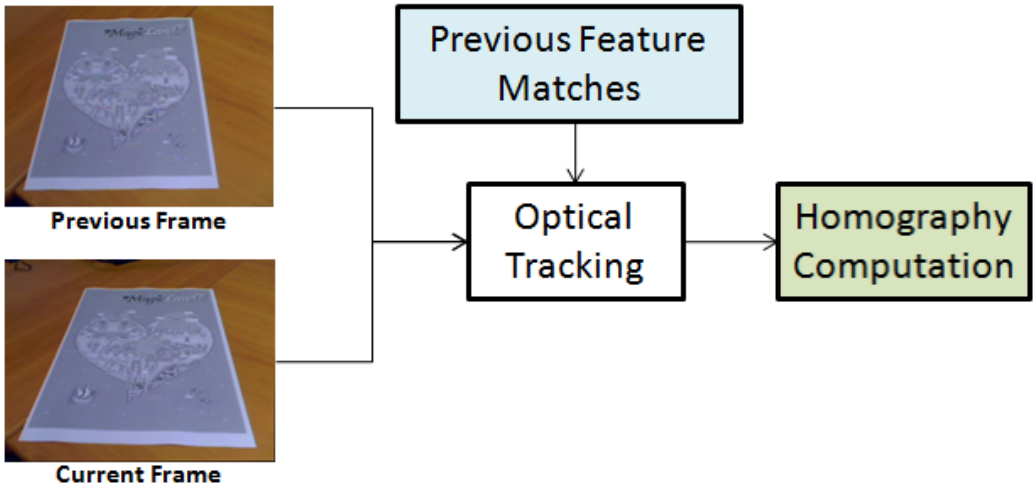


Figure 5.3: The optical flow method of natural feature registration. The previous image and matches are passed into the optical tracking module, as well as the current image. The optical tracking module returns the set of previous feature matches with the positions updated for the current image.

Planar registration is limited by perspective distortion due to camera transformation around the marker. Furthermore, natural feature registration is computationally intensive, which is undesirable for many registration applications. A solution to both problems is optical flow, the process of finding the transformation between two sets of data points over time. Optical flow can be added to the standard method of natural feature registration to track a previous found object across multiple frames.

As illustrated in Figure 5.3, the Optical Tracking step uses the current

frame, the previously registered frame and feature matches from the previous frame. The previous feature matches (shown highlighted in blue), are obtained from the feature matching stage of standard registration for the previous frame, as highlighted in blue in Figure 5.2. The Optical Tracking step attempts to locate the position of previous frame features in the current frame, based on motion between the previous and current frame as described in the following sections.

Optical flow involves following the motion of areas of constant brightness in an image, using the assumption that the brightness of an area does not change over time (Horn and Schunck 1981). This assumption is expressed as:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{5.1}$$

where I is the intensity of a point (x, y) at time t, and (dx, dy) is the transformation of the point at time dt, and is true for both quiescent points and those which are in motion. Although it is unrealistic that brightness will not change over time, at a real time frame rate the difference in brightness between two consecutive frames is minimal, and the assumption "works so well in practice" (Fleet and Weiss 2005).

To derive a two dimensional function for optical flow, first optical flow in one dimension is considered.

5.2.1 Optical Flow in One Dimension

A one dimensional signal $I(x, t)$ is translated by dx after time dt to form the signal $I(x + dx, t + dt)$, as shown in Figure 5.4. The brightness constancy assumption shown in Equation 5.1 is represented in one dimension as:

$$I(x, t) = I(x + dx, t + dt) \tag{5.2}$$

This is expanded using the Taylor series. Only the first-order is considered with the assumption this approximates the transformed image well (Fleet and Weiss 2005). The first-order Taylor series expansion of $I(x + dx, t + 1)$ about

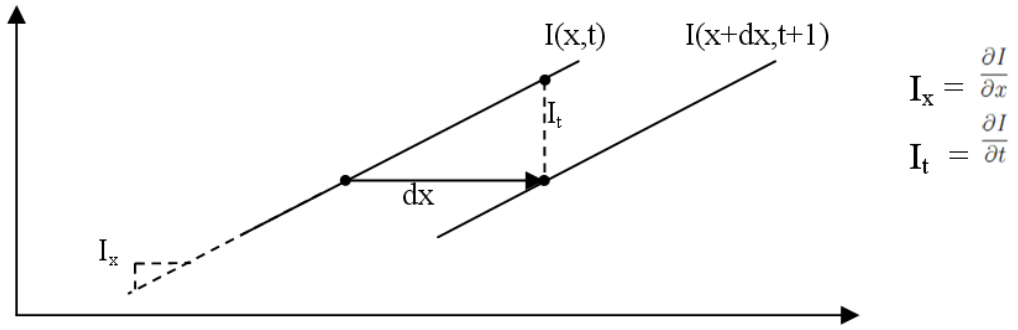


Figure 5.4: Optical flow in one dimension. The transformation dx of a one dimensional signal $I(x,t)$ after one iteration is equal to the temporal derivative I_t over the spatial derivative I_x

x is:

$$I(x+dx, t+dt) = I(x,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial t}dt \tag{5.3}$$

The temporal derivatives are represented as I_x and I_t:

$$I_x = \frac{\partial I}{\partial x} \quad I_t = \frac{\partial I}{\partial t} \tag{5.4}$$

Equation 5.2 is substituted into Equation 5.3 to form:

$$I(x+dx, t+dt) = I(x+dx, t+dt) + I_x dx + I_t dt \tag{5.5}$$

$$0 = I_x dx + I_t dt \tag{5.6}$$

The translation dx can be solved by rearrangement:

$$I_x dx = -I_t dt \tag{5.7}$$

$$dx = -\frac{I_t dt}{I_x} \tag{5.8}$$

In the example shown in Figure 5.4 $dt = 1$, and as the signal is linear

dx is equivalent to the temporal derivative over the spatial derivative. For non-linear signals, dx provides an approximation of the transformation.

5.2.2 Optical Flow in Two Dimensions

The equations for optical flow in one dimension can be generalised into two dimensions with the addition of the y dimension. Assuming the two dimensional brightness constancy assumption in Equation 5.1, the first-order Taylor series expansion of $I(x + dx, y + dy, t + dt)$ gives:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \tag{5.9}$$

Subtracting the brightness constancy assumption in Equation 5.1 from the Taylor series expansion in Equation 5.9 results in:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \tag{5.10}$$

$$\tag{5.11}$$

Which is then rearranged to form:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy \quad = \quad -\frac{\partial I}{\partial t}dt \tag{5.12}$$

$$\tag{5.13}$$

If the image derivatives I_x, I_y, I_t are defined as:

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y} \quad I_t = \frac{\partial I}{\partial t} \tag{5.14}$$

Then Equation 5.13 can be simplified to:

$$-I_t dt = I_x dx + I_y dy \tag{5.15}$$

Equation 5.15 is known as the "optical flow constraint equation", where

I_x is the partial spatial derivative of image I in the x dimension, I_y is the partial spatial derivative of image I in the y dimension, I_t is the partial temporal derivative of image I, and dx, dy is the transformation of the point at (x, y).

In one dimension, the transformation dx can be calculated, however in two dimensions, estimation from a single pixel is under constrained and will not result in a unique solution due to the two unknowns $I_x dx$ and $I_y dy$. To resolve this, a window of pixels can be tracked assuming that neighbouring points will all move in a similar motion, an assumption called "spatial coherence". Tracking a window of pixels will result in multiple linear equations which can be solved using least-squares minimization to find a unique result, as is used in the Lucas-Kanade optical flow method (Lucas and Kanade 1981).

For a window of $M \times M$ pixels p, where $N = M^2$, Equation 5.16 shows the linear system which is resolved to calculate flow :

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ ... & ... \\ I_x(p_N) & I_y(p_N) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ ... \\ I_t(p_N) \end{bmatrix} \tag{5.16}$$

The size of the window affects tracking accuracy; if the window is too large, then it is less likely that the spatial coherence assumption will be valid. If the window is too small, then the aperture problem can arise, where a lack of unique two dimensional information makes motion estimation impossible. The aperture problem can be avoided by only tracking features which have are distinguishable in two perpendicular directions, such as corners and points.

As only the first-order Taylor series is used, the initial estimate of the vector may not be accurate. Equation 5.9 is only an approximation of the optical flow and may not result in the optimal result due to noise and the brightness constancy assumption not being true. From this initial estimate of the optical flow vector, an optimal solution can be determined iteratively by adjusting the time derivative I_t, while keeping the spatial derivatives constant.

5.2.3 Optical Flow and Registration

Optical flow calculates the current position of known features from previous frames. By tracking the features found by the registration algorithm, the homography of the planar marker, and thus the camera extrinsic parameters, can be calculated without the need to perform registration again. Optical flow is far more efficient than natural feature registration as the calculations only require intensity based calculations on a sparse set of features. As optical flow is an iterative process, the windows surrounding features can change over time, resulting in a degree of perspective invariance.

Optical flow is susceptible to changes in illumination and noise due to the brightness constancy assumption. This results in "drift", the error in the calculated position of a feature increases over time, resulting in a decrease in the accuracy of the homography calculation over time. This is further complicated by features being discarded due to problems with noise, illumination, or large motion causing the features to fall outside of the search window. Failure due to large motions has been minimised in implementations such as pyramidal Lucas-Kanade optical flow (Bouguet 2002) which searches over multiple scale spaces to increase search time and allow a greater amount of movement between frames.

In this research, the Pyramidal implementation of the Lucas-Kanade optical flow method is used (Bouguet 2002). Although many other optical tracking algorithms have been developed, the Lucas-Kanade method is both fast and able to handle sparse point tracking, as opposed to only calculating dense flow fields like many other optical tracking algorithms. The initial results published by Baker, Roth, Scharstein, Black, Lewis and Szeliski (2007) for the Middlebury Optical Flow dataset[1] rate Pyramidal Lucas-Kanade as the third best of the five tested algorithms, and the current results show that it is still one of the faster optical flow algorithms.

Although this research uses the Pyramidal implementation of the Lucas-Kanade, any optical flow algorithm which has the capacity to handle sparse point tracking would be suitable, and could be easily implemented into the framework to replace Lucas-Kanade.

[1] `http://vision.middlebury.edu/flow/`

5.3 OPIRA method of Natural Feature Registration

OPIRA is a method of registration designed to improve invariance to changes in rotation, scale and perspective distortion for any natural feature registration algorithm. Once and initial registration has occurred, OPIRA iteratively refines the results of registration over multiple frames. Figure 5.5 shows a basic overview of how the OPIRA method.

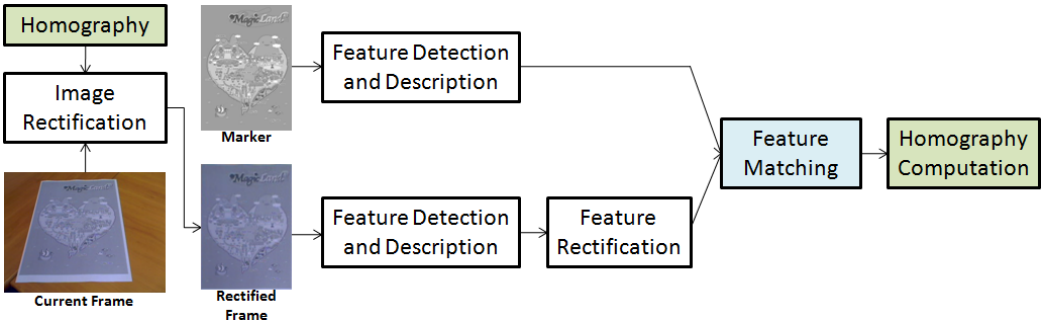


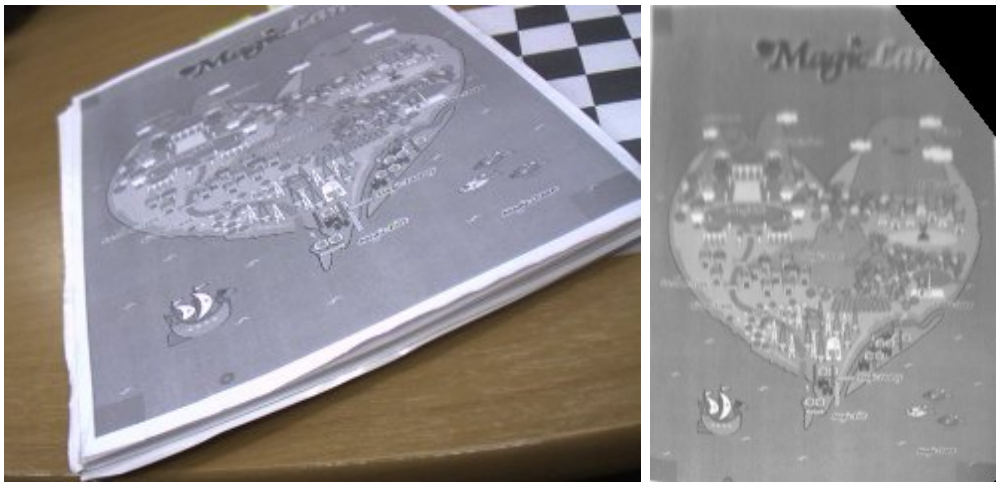
Figure 5.5: The OPIRA natural feature registration pipeline. In addition to normal registration, OPIRA uses image rectification to create a rectified image, which feature detection and description is performed on. Any features are rectified using the inverse of the homography and matched to generate a new feature match set for homography computation.

A homography (highlighted in green) obtained from optical flow or standard methods of registration (highlighted in green in Figures 5.2 and 5.3 respectively) is passed into the image rectification stage with the current frame. Image rectification is performed using backwards projection; for each pixel p to be found in the rectified frame, a corresponding sub-pixel p' is found in the current frame by multiplying the position of p by the inverse of the homography, calculated using SVD. Bilinear interpolation is performed over p's four nearest neighbour pixels P_{11}-P_{22} using Equation 5.17 to determine

the value of the corresponding pixel in the rectified frame.

$$\begin{aligned} P_{11} &= int(p'.x, p'.x) & P_{12} &= int(p'.x + 0.5, p'.x) \\ P_{21} &= int(p'.x, p'.x + 0.5) & P_{22} &= int(p'.x + 0.5, p'.x + 0.5) \\ p &= P_{11}(1-x)(1-y) + P_{21}x(1-y) + P_{12}(1-x)y + P_{22}xy \end{aligned} \tag{5.17}$$

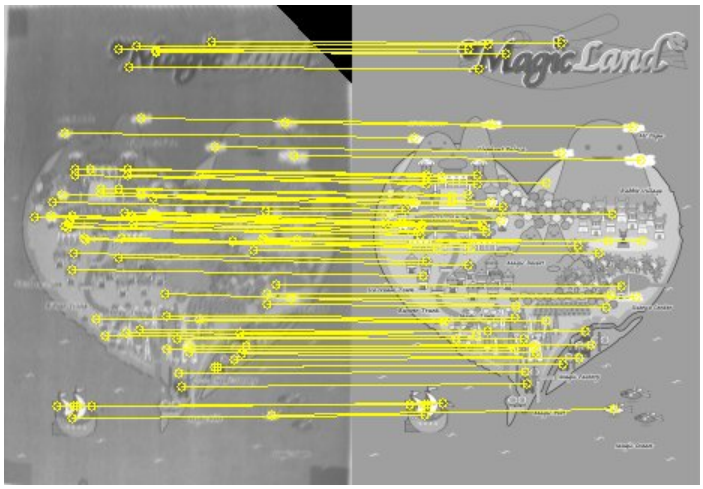
Once the value for each pixel in the rectified frame has been computed the result is a rectified image of the current frame which represents a mapping from the frame to the same scale, orientation and rotation of the marker, as shown in Figure 5.6. The accuracy of this depends on the accuracy of the initial homography, although even poor estimates usually result in a rectified frame which is more similar to the marker than the original frame.



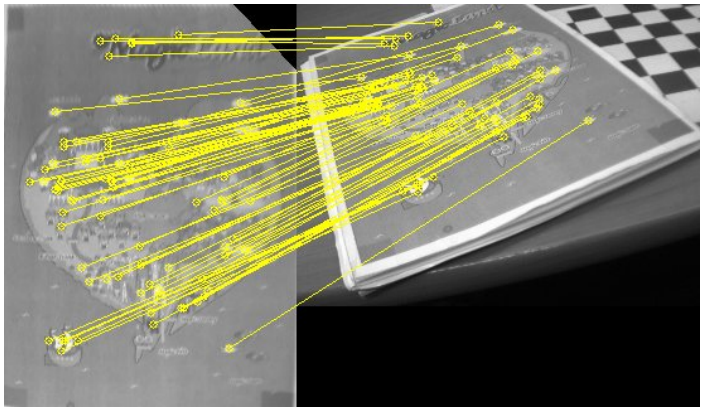
(a) (b)

Figure 5.6: Rectifying perspective distortion, (a) The source frame, (b) The perspective rectified marker found. Areas of the marker not visible are rendered as black so as to not introduce additional feature points.

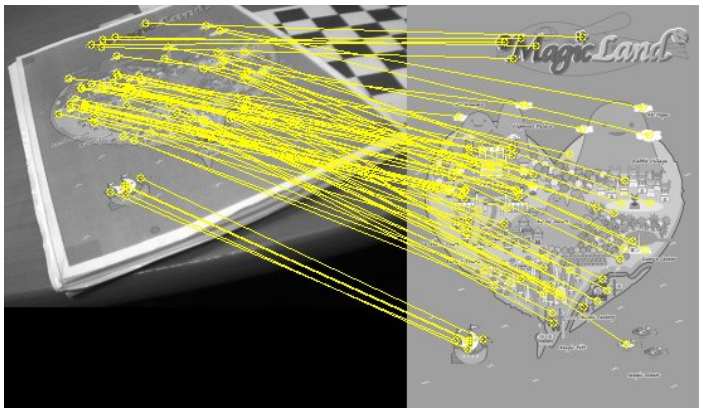
Once a rectified frame is obtained, the features are extracted and multiplied by the inverse of the homography to calculate their positions in the original frame. These features are then matched against feature from the marker image, and the results used to compute a homography. This process is illustrated in Figure 5.7.



(a)



(b)



(c)

Figure 5.7: Registration and rectification of matches with OPIRA, (a) Registration between the perspective rectified frame and marker, (b) Matches found in the perspective rectified frame and their positions in original frame based on the inverse homography, (c) Matches in original frame and marker, used for calculating the extrinsic camera parameters

The feature matches computed during the OPIRA stage (highlighted in blue) can be passed into the optical flow method (highlighted in blue in 5.3), so that the refined matches are then tracked into the next frame. Optionally, further refinement can be performed by passing the results of this homography computation back into OPIRA.

The rectified images obtained using OPIRA will often be less detailed and have more blur than the original marker image. This is an inherent problem with perspective distortion, as the marker rotates away from the camera, the area of the marker represented by a single pixel increases. When rectification is performed, areas of the marker further away from the camera will have greater blur and less detail than those closer to the marker, as can be seen in Figure 5.7(b), where the top of the marker is less detailed than the bottom. This problem is handled by the invariance to blur and reduced detail that is inherent in the natural feature registration algorithm used with the OPIRA method. Typically markers created for natural feature registration applications will be a high quality digital source, and the cameras used for registration will be significantly lower resolution and prone to noise. Because of this, a natural feature registration algorithm must have some invariance to blur and reduced detail in order for registration to succeed. OPIRA takes advantage of this to provide perspective invariance by transforming the problem of perspective distortion to one of low detailed images.

Unlike the affine invariant detectors discussed in Section 2.2.4 or the feature classifiers discussed in Section 2.2.5, OPIRA requires no additional training. Natural feature registration algorithms such as the Ferns Classifier (Ozuysal et al. 2009) require many minutes to train even low resolution images on a standard desktop computer, while OPIRA requires no additional training to provide similar or better levels of perspective invariance. This lowers both the time required for set up and the memory requirements of the system.

In the most robust implementation, OPIRA uses a "best of three" selection process to select the optimal feature match set from the standard method registration, optical flow method of registration, and registration of the rectified image. This results in a computational time of typically slightly over double the standard method, although the actual increase is determined

by the size of the marker. The compromise of this increased complexity is a large increase in robustness to perspective invariance and changes in rotation and scale.

In the OPIRA method, new features are constantly identified by the registration of the rectified image to replace those lost during optical flow or due to occlusion, and the standard method of registration is only used to re-establish tracking in the event of failure. By limiting the amount registration is run and reducing the robustness slightly, the computation speed of OPIRA can be increased considerably.

5.3.1 Fast-OPIRA

Fast-OPIRA provides the same functionality as OPIRA, but with a large increase in speed with only a minimal decrease in robustness. Instead of using the best of three selection process to choose the optimal feature match set, each matching strategy is assigned a threshold which is used to determine whether a given approach has identified enough matches to provide a reliable homography. If enough matches are not found, the next method in the hierarchy is executed. The hierarchy is arranged based on the operational cost of the approach, with the order being:

1. Optical flow

2. Registration of rectified image

3. Registration of original frame

The fastest but least robust method, optical flow tracking, is performed first. When the number of features tracked falls below an acceptable threshold, registration of the rectified image is performed to add more matches to the tracked feature set and control is returned to optical flow. During runtime these two approaches usually provide adequate feature matches for registration; however during initialisation or loss of tracking due to occlusion or fast movement, registration of the original frame is performed to establish a new feature set.

An additional improvement provided by Fast-OPIRA is the ability to use different registration algorithms for the rectified image and the original frame. As the rectified image is rotation and scale invariant the algorithm used does not need invariance to these conditions and a faster algorithm can be used. Registration of the original frame can use a more complex and slower algorithm, as it is seldom performed. This is useful for both SIFT (Lowe 2004) and SURF (Bay et al. 2006) registration algorithms, which have non rotation invariant implementations that are up to four times faster than the rotation invariant implementations.

Multi-Threaded code allows further speed gains, especially on increasing common multiple core computers. Registration of the rectified image is run in a thread on a secondary core, while an optical flow thread runs on the main processor and updates the transformation matrix with real time performance. Once the registration of the rectified image is complete, these results are then used for tracking in the optical flow thread, and registration begins again on the latest frame. Registration of the original image can be run in a third thread, or optionally only when the number of features found by optical flow and registration of the rectified image decreases below a threshold.

5.4 Summary

In this chapter, a new method of natural feature registration called OPIRA was presented, which reduces the limitations caused by changes in rotation, scale, and perspective distortion. The standard method of registration was discussed, and the optical flow method was introduced. Both methods were illustrated with a focus of how OPIRA can extend them and improve the results of registration. A less robust but more computationally efficient implementation called Fast-OPIRA was also presented, which uses a hierarchical approach, and optionally multi-threading, to provide much faster registration with only a minimal decrease in robustness. The OPIRA method is empirically evaluated in Chapter 7.

In the following chapter, solutions to the problems of noise and poor illumination are presented, and the effect of marker source on feature matching is discussed.

Chapter 6

Other Improvements for Natural Feature Registration

In Chapter 4 natural feature registration was decomposed into three main stages; feature detection, description and matching, and the common problems which negatively affect each stage were identified. In the first two stages, feature detection and description, there are five common image transformations and distortions which can reduce the effectiveness of registration, and in the final stage, feature matching, the source of the marker will affect registration accuracy.

In the previous chapter, a new method of natural feature registration called the Optical-flow Perspective Invariant Registration Augmentation (OPIRA) was proposed. OPIRA improves invariance to three of the five transformations and distortions; rotation, scale and perspective. In this chapter, methods to remove the remaining distortions; noise and poor illumination, are presented and the impact of marker source on registration is discussed.

6.1 Noise Invariance

The robustness and accuracy of registration is correlated with the quality of the source images. If the images are of low quality, the features detected will be less similar to those found in the marker, resulting in a decrease in valid matches and an increase in false positive matches. Assuming a properly calibrated camera, the largest problem is noise.

Noise is defined as a corruption of information when obtaining optical information with a camera and representing it digitally. The level of noise in an image is indicated by the Signal to Noise Ratio (SNR), a measure of the mean signal power of the light captured by the camera compared to the standard deviation of the noise power corrupting the image. As the global ambient illumination of a scene decreases, so too does the signal to noise

ratio, resulting in an increase in the effect of noise.

This research identifies two types of noise; local noise, which affects a localised part of an image, and global noise, which affects the image as a whole. Decreasing the illumination in a scene will increase in the level of local noise, such as Gaussian (Rank, Lendl and Unbehauen 1999) or Salt and Pepper noise (Li and Song 2004). As a result, many cameras automatically adjust their exposure time to maximise the light captured and reduce the level of local noise. With an increased exposure time, the effects of global noise, such as motion blur, increase, resulting in further corruption of the image.

Changes in the exposure time of a camera can be estimated by looking for periodic illumination changes between consecutive frames. These illumination changes are a result of hysteresis of the exposure time in response to changes in ambient illumination. This is done to reduce "flickering" caused by minor illumination fluctuations (Clark, Green and Grant 2007). The hysteresis can be observed as a "stepping" effect in the global brightness between the difference of the previous and current frame, as shown in Figure 6.1.

As both local and global noise increases with a decrease in ambient light, noise can be unified as an inverse function of global ambience (Clark and Green 2006). The level of both global and local noise in an image can be estimated with a measure of ambient light levels, however methods of noise removal are specific to the type of noise. The following sections discuss the two classes; local noise, the effect of which is not correlated to the position in the image, and global noise, the effect of which is spatially related to the image.

6.1.1 Local Noise

Local noise is characterised by small regions of corruption with no spatial correlation, and appear as "speckles" in the image. From a human perspective local noise is a minor problem, however the corrupted areas are often significantly distinct in their surroundings, making them probable candidates for feature detection.

The decreased signal in areas of low illumination means that local noise

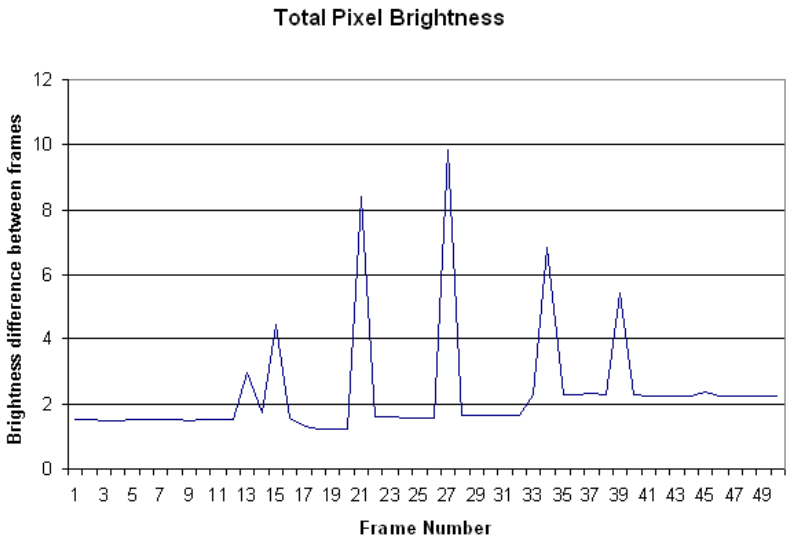
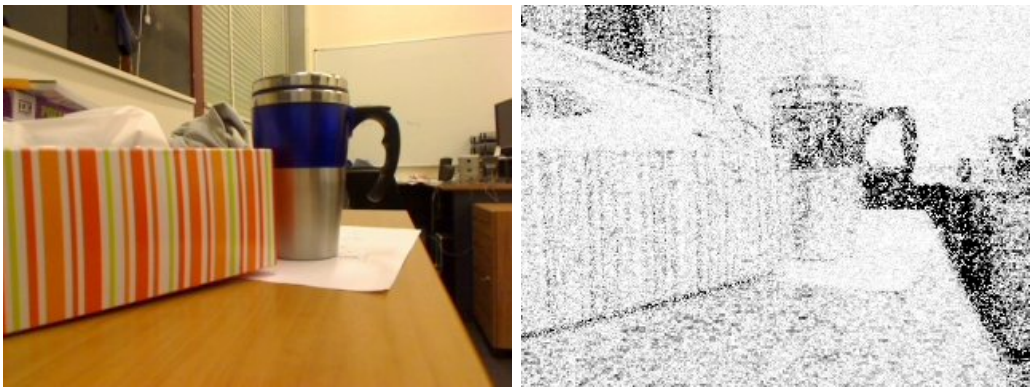


Figure 6.1: Periodic changes in exposure time appear as a "stepping" effect in the difference of two consecutive frames. Instead of a smooth exposure transition when global ambience changes, large jumps are made at regular intervals (Clark et al. 2007)

is more prevalent in darker areas of the scene. Figure 6.2(a) shows an image captured by a digital camera, and (b) shows the inverse of the difference image between two sequential frames scaled by intensity. The darker the pixel, the larger the difference between to two identical pixels between frames.

Local noise such as salt and pepper and impulse noise can be removed successfully with median filters (Chinnasarn, Rangsanseri and Thitimajshima 1998), although this can also remove legitimate features. To resolve this, a detail preserving regularization (Chan, Ho and Nikolova 2005) median filter can be used, or the application of the filter can be limited to areas of the image likely to contain high local noise content. The noise content of a region can be estimated by modelling the variance of pixels in the camera (Grant, Green and Clark 2007), as shown in Figure 6.3. These models can be generated during camera calibration.

Local noise is generally less detrimental to registration than global noise. Local noise is likely to result in more erroneous features being found by



(a) (b)

Figure 6.2: Gaussian noise present during camera capture, (a) A frame of input video, (b) The noise present between two frames of video. The camera was stationary, and frame was subtracted from the previous frame, and the result divided by the intensity. To make the noise more visible, (b) was multiplied by ten. The brighter the point in the original scene, the lower the noise.

the registration algorithm, and these features are generally identified and removed during the feature matching stage. Conversely, global noise such as blur degrades the entire image, and lowers the correlation between legitimate features in the frame with those in marker, resulting in less matches and subsequently less accurate registration.

6.1.2 Global Noise

Global noise is defined as corruption of the image as a whole, the deformation of any given point is spatially related to the deformation of neighbouring points. The most common forms of global noise are motion and out-of-focus blur. Motion blur is caused by motion during the exposure time of the camera while out-of-focus blur is caused by a defocused lens.

Natural feature registration achieves maximum robustness and accuracy with sharp images. Motion and out-of-focus blurs reduce fine detail, which results in a reduction of registration accuracy and robustness. Both types of blur are represented using the same mathematical function, as shown in

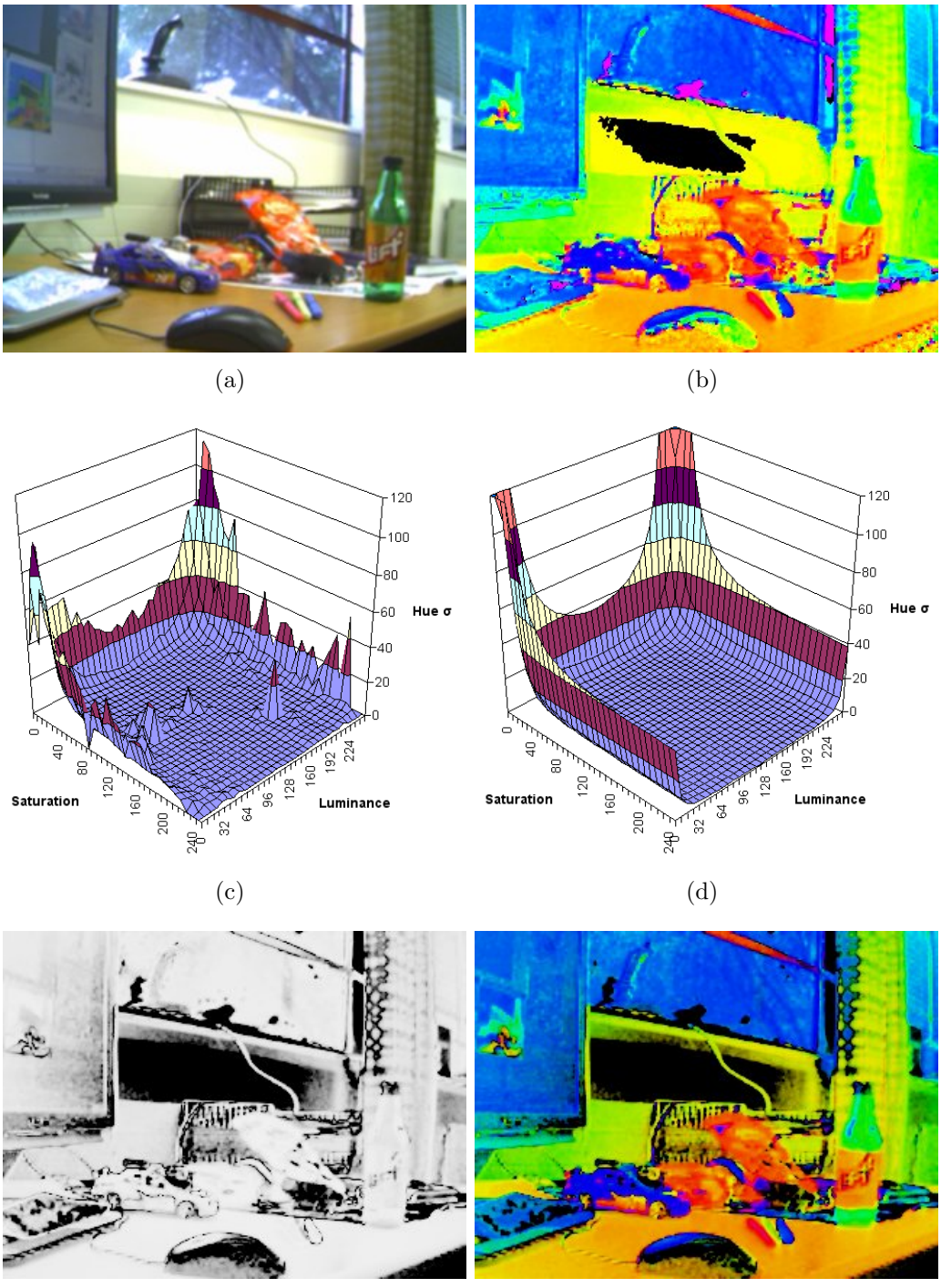


Figure 6.3: Estimation of the accuracy of pixel hue, (a) The original scene, (b) The hue component for each pixel, (c) Experimental data of hue variance compared to luminance and saturation, (d) Model fitted to c, (e) Image showing accuracy estimation for each pixel, (f) pixel hue with intensity saturation representing accuracy (Grant et al. 2007)

Equation 6.1, where i is the ideal image and g is the blurred image, h is the Point Spread Function (PSF) which defines how the light from a pixel is distributed, and the n function represents noise present in the image.

$$g(x, y) = i(x, y) \times h(x, y) + n(x, y) \tag{6.1}$$

Mathematically the method of removal for both blurs is the same, the only difference is the PSF used. To remove the problem of blur, the image is convolved with the inverse of the PSF. The PSF can be estimated using a blind deconvolution (Stockham, Cannon and Ingebretsen 1975) filter such as the Wiener filter (Biemond, Lagendijk and Mersereau 1990), which attempts to calculate the PSF which best fits the spectral content of the image, or the Richardson–Lucy algorithm (Richardson 1972), which assumes a Poisson noise distribution. Any noise present in the blurred image will increase the difficulty of obtaining the true point spread function. To avoid non optimal solutions and increase the speed of solution convergence, the blind deconvolution algorithms can be initialised with an estimate of the PSF.

For the purposes of registration, the non-iterative implementation of Wiener filter is ideal due to the simplicity and speed of the algorithm. As the implementation is non-iterative it cannot estimate the point spread function, and so the PSF must be calculated off-line. Out-of-focus blur is static and implicit to the camera, and the PSF can be calculated at camera calibration time, as described in the following section. The motion blur PSF can be estimated from global optical flow vectors if the exposure time is known, as described in Section 6.1.2.

The non-iterative Wiener filter algorithm is described in Algorithm 6.1.1. The image to be processed g and point spread function h are both scaled to the largest dimensions of the two images. The scaling function replicates the exterior rows and columns. The Discrete Fourier Transforms (DFT) of the image and PSF are found, and are used to calculate the real and imaginary spectra of the output image. The calculation involves a defined threshold *thresh*, *Gamma* value, and Signal to Noise Ratio *SNR*. The filtered image is found by computing the DFT of the calculated real and imaginary spectra.

The quality of the image obtained depends on the accuracy of the PSF

Algorithm 6.1.1: NON-ITERATIVE WIENER FILTER(g, h)

$width = max(g.width, h.width)$
$height = max(g.height, h.height)$
$enlargeImage(g, size(width, height))$
$enlargeImage(h, size(width, height))$
$DFT(g, gReal, gImag)$
$DFT(h, hReal, hImag)$
for each $pixel$ **in** g
  $Sh = max(hReal^2_{pixel} + hImag^2_{pixel}, thresh)$
  $Sf = gReal^2_{pixel} + gImag^2_{pixel}$
  $Sm = Sf/(Sh \times Sf + Gamma \times SNR)$
  $gReal_{pixel} = Sm \times (hReal_{pixel} \times gReal_{pixel} +$
    $hImag_{pixel} \times gImag_{pixel})$
  $gImag_{pixel} = Sm \times (hReal_{pixel} \times gImag_{pixel} -$
    $hImag_{pixel} \times gReal_{pixel})$
next
$DFT(gReal, gImag, g)$

estimate. The methods for estimating the PSF is different for out-of-focus blur and motion blur, which are described in the following sections.

Out-of-Focus Blur PSF estimation

Out-of-focus blur occurs when the distance between the lens and the sensor is not equal to the focal length f of the lens, as described in Section 3.2.2. With a perfect lens, the point will be radially blurred, and the PSF can be represented as shown in Equation 6.2, where the blur radius R is proportional to the distance between where the light is focused and the focal length f. Any deformations in the lens which are not removed as discussed in Section 3.3.2 will result in a non radial PSF. The out-of-focus PSF captured from a camera shown in Figure 6.4 shows a slightly non radial distribution of light.

$$h(x, y) = \begin{cases} 1/\pi R^2 & \text{when} \quad x^2 + y^2 \leq R^2 \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

The point spread function for out-of-focus blur can be estimated during

Figure 6.4: The slightly non radial distribution of light of an out-of-focus point spread function

camera calibration, as the PSF is static and intrinsic to the focal distance of the lens. The naïve method of obtaining the PSF is to capture an image of a single point light source at infinite distance with the camera. As this method is impractical, the PSF can instead be estimated by analysing the cepstral components of the blurred image (Childers, Skinner and Kemerait 1977). Out-of-focus blur appears in the cepstral domain as a circle of negative values (Stockham et al. 1975), the radius R in Equation 6.2 can be found by measuring the radius of the circle when the cepstral representation is converted to polar co-ordinates (Fabian and Malah 1991).

Motion Blur PSF estimation

Motion blur occurs when motion of the camera or scene occurs during the exposure time of the camera. The problem is increased when the exposure time increases because of a decrease in global ambience. The motion blur PSF is a rectangular shape with its major axis aligned in the direction of motion, and the length proportional to the velocity difference between the camera and the object during exposure. It is represented in Equation 6.3, where l is the length, and θ is the direction.

$$h(x,y) = \begin{cases} 1/l & \text{when} \quad -cos\theta \times l/2 \leq x \leq cos\theta \times l/2 \\ & \text{and} \quad -sin\theta \times l/2 \leq y \leq sin\theta \times l/2 \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

The motion blur PSF can be found in the cepstral domain as adjacent zeros at distance $\frac{1}{l}$ (Fabian and Malah 1991). Alternately, the direction θ and length l can be found by computing all motion vectors between the previous and current frame using optical flow. Once any outlier vectors are

removed, the direction is the average direction of all remaining vectors *vect*, and the distance is the average distance of all remaining vectors multiplied by a constant c over the exposure time t of the camera, as shown in Equation 6.5. The constant c is calculated during calibration by locating a point against a uniform background in two adjacent frames where motion has occurred, and measuring the length of blur in the direction of motion.

$$\theta = \frac{\sum_1^n (vect.dir)}{n} \tag{6.4}$$

$$l = \frac{\sum_1^n (vect.dist)}{n} \times \frac{c}{t} \tag{6.5}$$

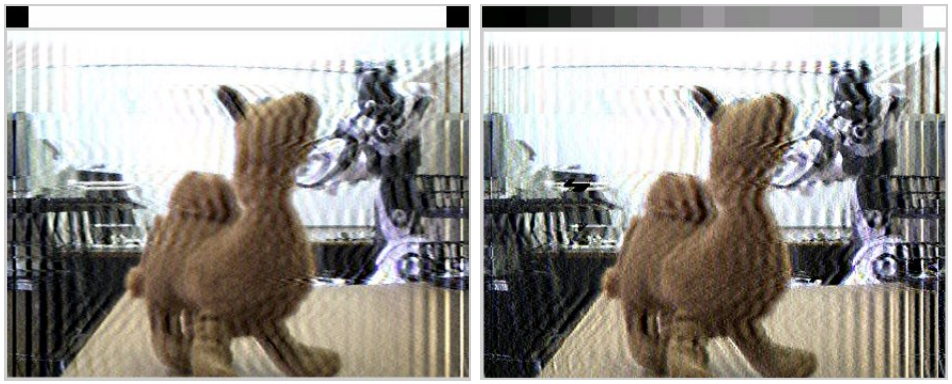
Figure 6.5 shows an example of motion blur PSF estimation and blur removal (Clark and Green 2006). Figure 6.5(a) shows the image sequence created by rotating the camera around its Y axis. Figure 6.5(b) is the third frame and (c) shows the optical flow vectors calculated between the second and third frames. The vectors are colour coded based on their orientations within 10° increments. Outlier vectors greater than one standard deviation from the mode of all orientation are eliminated. The PSF computed from inlier vectors is shown at the top (d), with the recovered image using Wiener filter deconvolution below. There is considerable "rippling" distortion within the image, which can be resolved by limiting the application of the filter to areas not near the borders of the image or around object edges (Jin, Fieguth, Winger and Jernigan 2003).

Figures 6.5(e) and (f) are comparison PSFs and recovered images found using a blind deconvolution implementation of the Wiener filter. Figure 6.5(e) was obtained using a good initial estimate of the PSF, and shows little difference in the recovered image compared with the estimated PSF. Figure 6.5(f) was obtained using a poor initial estimate of the PSF, which resulted in a non-optimal PSF and a very grainy and noisy recovered image. The computational time of the blind deconvolution Wiener filter was considerably greater than the non iterative Wiener filter.

The improvements gained by using deconvolution to reduce the problem of blur diminish as the level of blur increases. Eventually a point is reached

(a)

(b) (c)



(d) (e)

(f)

Figure 6.5: Motion blur removal from an video sequence, (a) the video sequence, (b) the third frame in the image sequence, (c) Optical flow vectors between frame two and third, (d-f) the estimated PSF (top) and deconvoluted image (bottom) for optical flow (d), good initial estimate blind deconvolution (e), and poor initial estimate blind deconvolution (f) (Clark and Green 2006)

where the image is too corrupted and the noise introduced during the deconvolution process only reduces registration accuracy further.

6.2 Illumination invariance

Many natural feature registration algorithms use illumination information in the calculation of feature descriptors. In environments with low ambient light, the difference in illumination for a marker is minimal, which can result in non robust feature descriptors. Poorly illuminated images can be corrected while retaining their integrity using contrast enhancement algorithms.

A common method of contrast enhancement is histogram equalisation as it is computationally efficient, parameterless and is an invertible operation (Russ 2002). As histogram equalisation increases the level of noise as well as valid data, it is only useful when a registration algorithm is more robust to noise than it is low-contrast images. Figure 6.6 shows a poorly illuminated image before and after histogram equalisation (top), and the histograms of the images before and after equalisation (bottom).

In order to perform histogram equalisation, the histogram of intensities of the image is found. A cumulative distribution function (CDF) is calculated by finding the accumulation of all entries in the histogram prior to the current CDF index. Each value in the CDF is normalized by multiplication with the maximum value (usually 255) divided by the number of pixels in the image. The normalised CDF is used as a intensity lookup to modify all pixels in the image. The histogram equalisation algorithm is shown in Algorithm 6.2.1.

With the limitations caused by the five problematic image transformations and distortions minimised, the feature detection and description stages of natural feature registration are more robust. To maximise accuracy during the feature matching stage, there must be a high correlation between the features of the marker and the frame. This correlation can be reduced when the marker and frame originate from different sources. This is discussed in the following section.

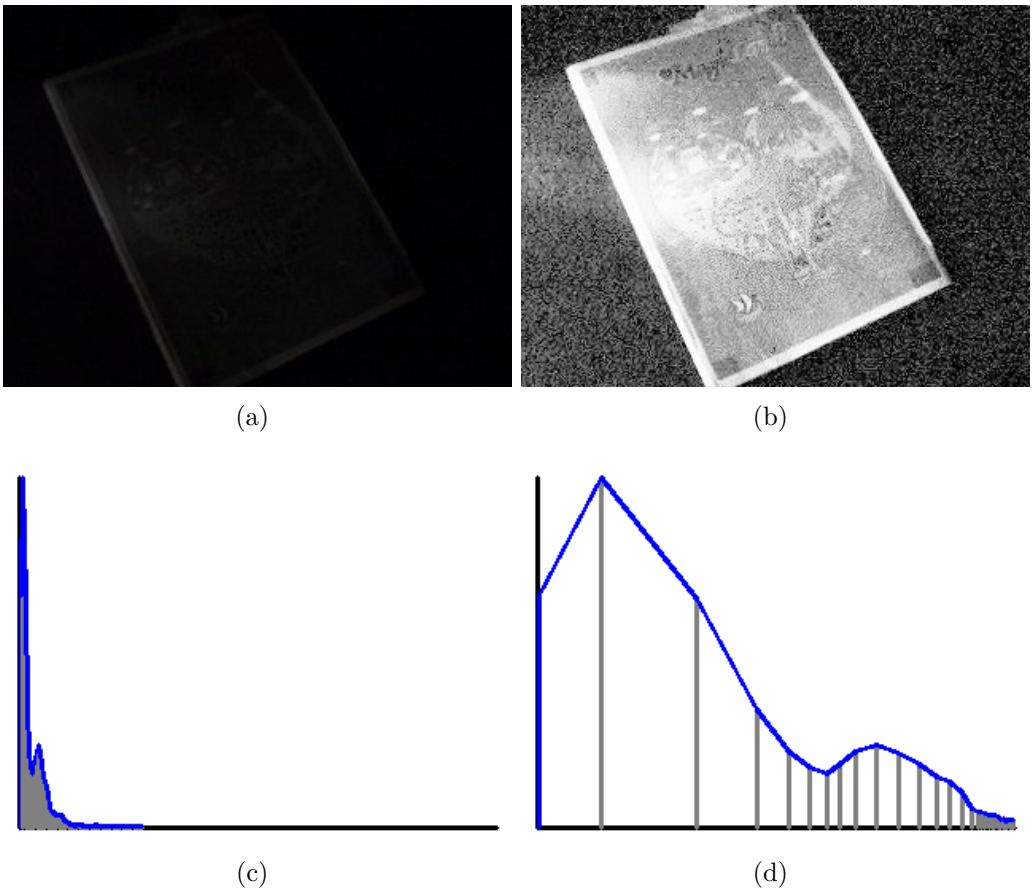


Figure 6.6: Histogram equalisation for contrast enhancement, (a) The marker in a poorly lit environment, (b) The image after histogram equalisation, (c) The histogram of (a), (d) The histogram of (b)

6.3 Marker Sources

The robustness and accuracy of registration depends on the similarity between the features found in the marker training image and the corresponding features found in each frame of the video stream. When the marker and frame have not been captured by the same device, this similarity is reduced.

There are two sources which can be used for the marker training. The first is the high quality digitally created source, such as an image captured by a scanner or created on a computer. With a high quality source image, the detrimental effects of noise and deformations of the marker image are

Algorithm 6.2.1: HISTOGRAM EQUALISATION(im)

$hist[256]; cdf[256];$
for each $pixel$ **in** im
  $hist[pixel.value]++;$
next

$cdf[0] = hist[0];$
for $i = 1$ **to** 255
  $cdf[i] = cdf[i-1] + hist[i];$
next

for $i = 0$ **to** 255
  $cdf[i] = cdf[i] \times \frac{255}{im.width \times im.height}$
next

for each $pixel$ **in** im
  $pixel.value = cdf[pixel.value]$
next

minimised. This guarantees that features trained in the marker are free from noise, and ensures a high level of detail for training features at high scale spaces. The success of registration in this instance relies on a high quality production of the physical marker from the source material, and high quality images received from the camera. If either the physical marker or images from the camera are poor quality, the difference between the features trained in the marker and those found in the video may be too great for successful registration.

If a high quality marker cannot be created, or the camera is of poor quality, the second option is to capture an image of the marker with the camera, and use this image as the source for training. The marker will not be as clear or highly detailed as a high quality digitally created source, and any noise introduced by the camera will be included in the features created during the training, however the features trained in the marker will be more similar to those captured during registration. The properties unique to the camera,

(a) (b)

Figure 6.7: The same marker from two different sources, (a) The original digital source, (b) The marker as captured by the camera

such as distortion and defocusing, will be present in the training image. If the marker is captured in the same environmental conditions as it is used in, the lighting characteristics found in the marker will match those in the frame, resulting in a higher correlation between corresponding features. For optimal results, each camera used and registration environment will require reacquisition of the marker image, increasing the calibration workload but increasing the robustness of the system.

Figure 6.7 shows the difference between the high quality source image (which has been downsized) and the image as it is seen by a camera. The lighting characteristics are considerably different and the image captured by the camera is defocused. Due to lens distortion the aspect ratio is different between the two images.

6.4 Summary

In the Chapter 4, five image transformations and distortions which limit the accuracy of the feature detection and description stages were identified. In the previous chapter, the effects of three of these transformations and distortions were discussed, and the OPIRA algorithm presented which improved the robustness of registration algorithms to these factors. In this chapter, the remaining two distortions; noise and poor illumination were presented, and solutions to their effects were discussed. Additionally, the effects of marker source on the success of the feature matching stage were discussed.

The following chapter empirically evaluates the OPIRA method, the noise and poor illumination solutions, and the effect of marker source on feature matching.

Chapter 7

Evaluation

In the previous chapters, image transformations and deformations which reduce the accuracy of the feature detection and description stages of natural feature registration were investigated, and methods to overcome these limitations were described. In addition the effect of marker source on feature matching was also discussed.

Of particular importance, a new registration method called OPIRA was presented. OPIRA was designed to reduce the impact of changes in scale, rotation and perspective distortion for natural feature registration algorithms.

This chapter empirically evaluates each of the methods designed to improve feature detection and description, and the effect of marker source on feature matching. The experimental environment in which the evaluations were conducted is described, including the software framework which was developed. The implementation and calibration of an external ground truth is discussed and each proposed improvement is evaluated.

7.1 Experimental Setup

To ensure the evaluations were consistent and repeatable, a uniform test environment was established. This setup included the Camera, Registration Framework, Measures of Accuracy, Markers and the Evaluation Data. Each of these components was controlled across all experiments, resulting in a baseline ground truth which ensures any unknown variables are consistent for each evaluation.

All development and evaluations were conducted on a 32 bit, 2.4GHz dual core Intel computer with 2GB RAM running Windows XP. The room and lighting conditions are described in Section 7.1.5.

7.1.1 Camera

In this research, potential weaknesses in the natural feature registration process are identified and solutions proposed and evaluated. Noise can be reduced by the use of expensive state of the art equipment, such as high quality cameras, and by carefully controlling the lighting and environment. For a many natural feature registration applications, these restrictions are not viable. Any application designed for the mass market needs to be able to function on a variety of low quality cameras in unconstrained environments. With this motivation, the evaluations were conducted using a typical consumer quality camera.

The camera chosen for the evaluations was an ADS USB2.0 Turbo WebCam[1], shown in Figure 7.1. This camera supports up to 640×480 resolution at 30 Frames Per Second (fps) uncompressed video. In practice, natural feature registration at any resolution over 320×240 was too slow to be considered real time on a typical desktop computer.

Figure 7.1: The camera used for experimental evaluations, an ADS USB2.0 Turbo WebCam[1]

[1] `http://www.adstech.com/Support/ProductSupport.asp?productId=usbx2020`

The camera features light sensitivity to 1.7V/lux, automatic gain control and white balancing, a manual focus lens with focal length of 6.0mm and 52° field of view. Video capture was supported using standard WDM device drivers, and all camera parameters set to their defaults. The camera has a tripod mount, which proved useful to attach the camera to the test rig described in Section 7.1.5.

7.1.2 Registration Framework

An object oriented software framework was constructed to allow automation of the evaluations over a range of registration algorithms, using the OpenCV library (Bradski 2000) for image capture and low level computer vision algorithms. The software was developed in C++ in the Visual Studio 2005 environment.

In the framework, the natural feature registration process is encapsulated in a `Registration` class, as shown in Figure 7.2. The `Registration` class is instantiated with the marker image and registration algorithm to use during the evaluation. The `performRegistration` function performs registration between the marker and the current video frame, using the intrinsic camera parameters to calculate the extrinsic camera parameters.

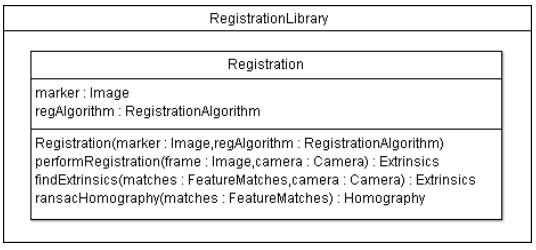


Figure 7.2: The `Registration` class, which uses a `RegistrationAlgorithm` to perform registration on a `frame`, and returns the camera extrinsic parameters

Once a set of `FeatureMatches` has been identified between the marker and the video frame, the `ransacHomography` function is called to remove outliers and calculate the homography. The separation of the homography

and extrinsic camera parameter calculations from the registration algorithms isolates any differences in the results to the registration algorithms.

Registration Algorithms

Three popular natural feature registration algorithms were used for the experimental evaluation of the solutions proposed in Chapter 5:

1. SIFT (Lowe 2004), as described in Section 2.2.2, was implemented using source code available from the VLFeat library[2]. The scale and octave levels were both set to three, with the first scale space computation at the original image dimensions. These parameters were found experimentally to provide the optimal compromise between robustness and speed.

2. SURF (Bay et al. 2006), as described in Section 2.2.3, was implemented by designing an interface to the author's own libraries, available from their website[3]. The parameters used in the evaluations were the same as in the author's code.

3. The Ferns classifier (Ozuysal et al. 2009), as described in Section 2.2.5, was implemented using the author's own source code, available from their website[4]. Several corrections were made to the source code and submitted to the authors of Ferns. The parameters used in the evaluations were the same as the author's code. For conciseness, the Ferns classifier is referred to simply as Ferns.

The aim of the evaluations in this chapter is not to compare the accuracy between registration algorithms, but to evaluate the improvements when using the solutions discussed in the previous chapter. The rotation invariant implementations of the SIFT and SURF algorithms were significantly slower than the rotation dependent algorithms, making them unsuitable for

[2] `http://www.vlfeat.org/`

[3] `http://www.vision.ee.ethz.ch/~surf/download.html`

[4] `http://cvlab.epfl.ch/software/ferns/index.php`

real time registration. For these reasons, the rotation dependent SIFT and SURF algorithms are used for most experiments, while the rotation invariant algorithms are evaluated in Section 7.2.3 to ensure a fair comparison.

Figure 7.3 shows a class diagram of the implementation of the evaluated registration algorithms. All the registration algorithms derive from the generic interface, `RegistrationAlgorithm`, which defines instantiation with the marker image, and the function `findMatches` which finds feature matches in the marker and current frame.

An approximate nearest neighbour approach was implemented for feature matching, encapsulated by the `RegistrationAlgorithmANN` abstract class. The ANN is constructed on instantiation, and features found in the marker are inserted. The Ferns algorithm requires a special classifier, and thus does not inherit from this class.

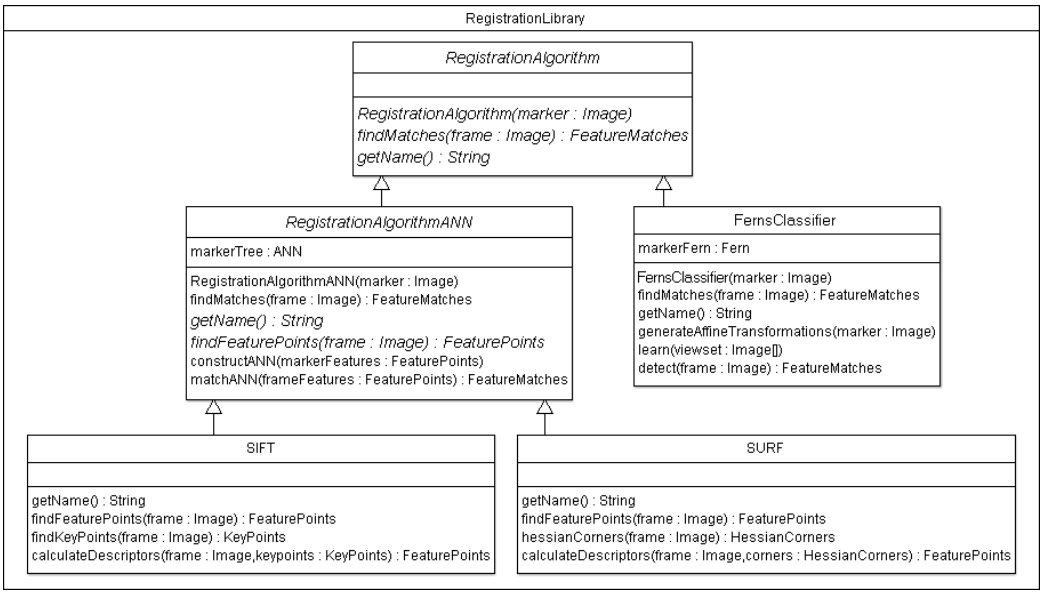


Figure 7.3: The RegistrationLibrary package, which implements a number of registration algorithms with a generic interface, `RegistrationAlgorithm`

Registration Methods

The `Registration` class shown in Figure 7.2 was extended to evaluate the effectiveness of OPIRA compared to standard registration and registration with optical flow. The `RegistrationOpticalFlow` class shown in Figure 7.4 stores the previous frame and matches, which are used for tracking in the `performOpticalFlow` function. To maximise robustness, registration is also performed on each frame, and the method with the maximum matches after RANSAC is used to calculate the extrinsic camera parameters.

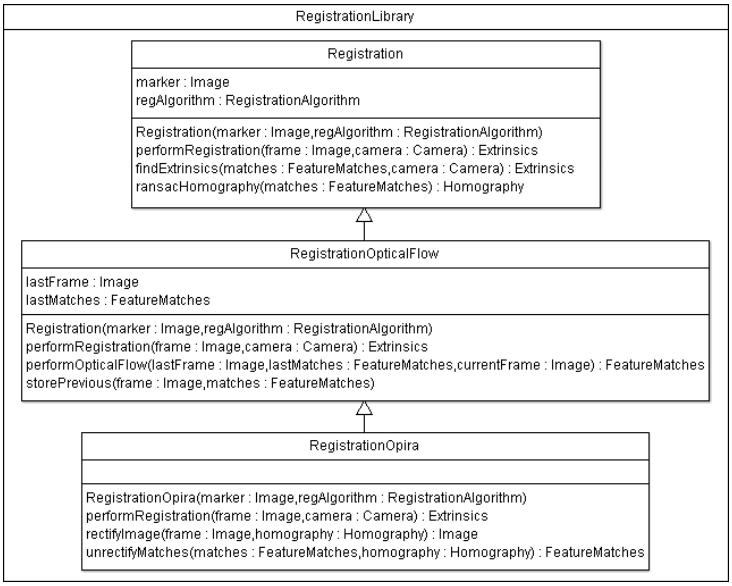


Figure 7.4: The registration and optical tracking classes `RegistrationOpticalFlow` and `RegistrationOpira`.

The OPIRA method, described in Chapter 5, is implemented in the `RegistrationOpira` class. Optical flow and registration are performed on each frame, and the homography of the method with the maximum matches after RANSAC is used to rectify the image. Registration is performed on the rectified image, and if the number of matches after RANSAC exceeds the other two methods, the matches are converted to the original frame with the `unrectifyMatches` function.

7.1.3 Measures of Accuracy

To evaluate the improvements in registration accuracy as a result of the improvements discussed in previous chapters, methods of measuring the accuracy must be defined. In the context of certain applications, such as augmented reality, qualitative measures such as visual inspection may be sufficient. Unfortunately these measures cannot be used for more precise quantification of registration accuracy because the granularity is low and the results are subjective to the observer, and should only be used for preliminary evaluation.

The evaluations used in prior research compare the repeatability (Grabner et al. 2006) or recall (Mikolajczyk and Schmid 2005) of specific features in an image, without the use of an external ground truth. While these measures do indicate the accuracy of registration, they do not evaluate the overall accuracy of the final registration transformation. The evaluation used in this research aims to evaluate this overall accuracy, as it is this measure which determines the accuracy of the registration in an application and ultimately affects the end users experience.

In this research an inertial orientation sensor is used as an external ground truth (discussed in Section 7.1.5) which provides the three rotational angles of the camera. To compare between the ground truth and calculated registration matrix, the registration matrix is decomposed into the three rotations using Shoemake's (1994) algorithm. The rotation system chosen was a body-aligned yxz, or yxz_r system, as this aligned the inertial tracker's pitch, yaw and roll with rotations about the x, y and z axes of the camera respectively.

Three quantitative measures of accuracy were used in this research:

1. Mean Absolute Error (MAE). The Mean Absolute Error is the average difference of the camera's rotation when calculated using registration and when measured using the ground truth, as shown in Equation 7.1, where $r_{x,y,z}$ are the x, y and z rotation angles calculated using registration, and $i_{x,y,z}$ are the x, y and z rotation angles measured using the inertial groundtruth. Mean Absolute Error was chosen over Mean Squared Error as the results are less skewed by the presence of outlier results common when erroneous transformations are calculated. Error

can only be calculated for frames where registration provided enough features to calculate a transformation.

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |r_{x,j} - i_{x,j}| + |r_{y,j} - i_{y,j}| + |r_{z,j} - i_{z,j}| \tag{7.1}$$

2. Average number of feature matches. As described in Section 3.3.1, the homography of a marker is computed using least squares minimization. By increasing the number of accurate feature matches, the impact of noisy or erroneous feature matches is decreased, resulting in a more stable calculated transformation. The feature matches are counted after RANSAC has removed outlier features.

3. Percentage of successfully registered frames. In this research, registration is considered successful if the maximum absolute difference of the camera's rotations when calculated using registration and when measured using the ground truth is falls below a defined threshold. This tolerance was experimentally set to a maximum of 5° rotation in any axis, as shown in Equation 7.2, where F is the set of all frames, $r_{x,y,z}$ are the x, y and z rotation angles calculated using registration, and $i_{x,y,z}$ are the x, y and z rotation angles measured using the inertial groundtruth. A visual representation of the maximum error accepted in a successful registration is shown in Figure 7.5.

$$\% = \frac{\sigma max(|r_x - i_x|, |r_y - i_y|, |r_z - i_z|) \leq 5^{(F)}}{F} \times 100 \tag{7.2}$$

Although the computational time of the proposed solutions is important when used in a real-time application, this research only focuses on the improvements to registration accuracy.

7.1.4 Markers

All evaluations were conducted over a range of markers to thoroughly assess the performance of all proposed improvements. The initial marker used was the "MagicLand" marker from the ARToolKit NFT, as shown in Figure 2.11,

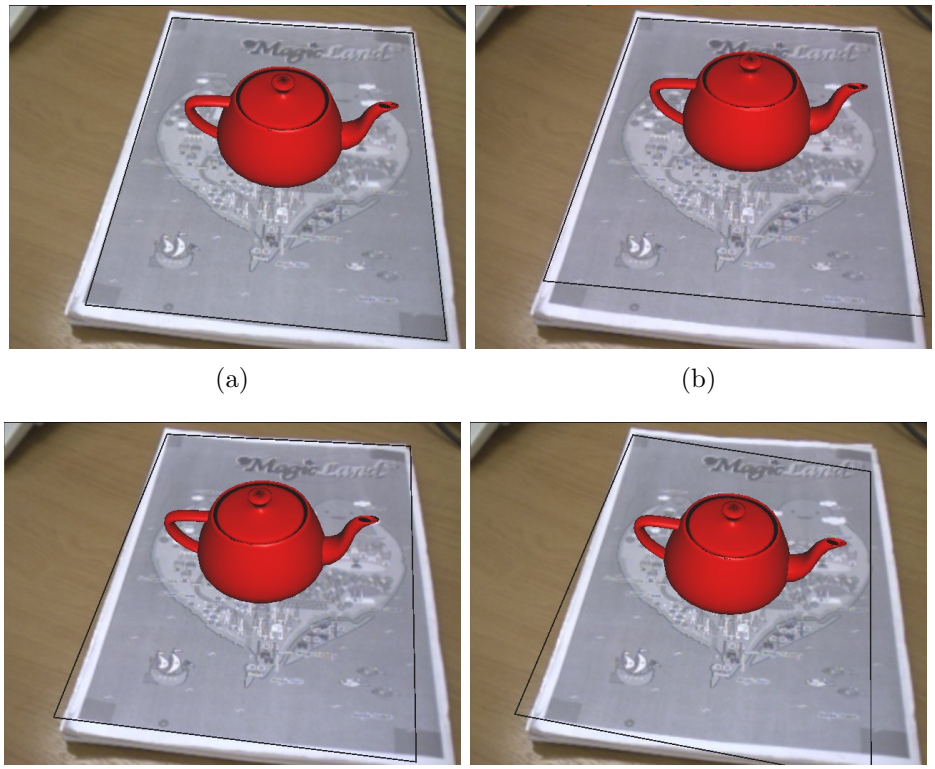
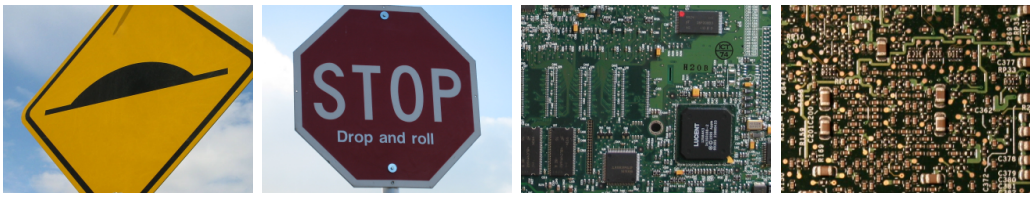


Figure 7.5: The maximum threshold for acceptable misalignment, (a) optimal alignment, (b) 5 ° misalignment in X axis, (c) 5 ° misalignment in Y axis, (d) 5 ° misalignment in Z axis

with the fiducial initialisation marker removed. In addition, the markers used by (Lieberknecht, Benhimane, Meier and Navab 2009) for their evaluation of template tracking algorithms were tested. These markers were chosen as they represent the four main categories of markers, those with Low texture, Repetitive texture, Normal texture and High texture. The markers are shown in Figure 7.6.

For conciseness, experimental graphs are only shown for the MagicLand marker, while the results of the experiments for all markers are tabulated at the end of each experiment.

Experimentally it was found that none of the implementations of any registration algorithm were able reliably track Lieberknecht et al.'s (2009)



(a) Bump (b) Stop (c) Lucent (d) MacMini

(e) Isetta (f) Philadelphia (g) Grass (h) Wall

Figure 7.6: The additional markers used in the evaluation process: (a-b) Low Texture, (c-d) Repetitive Texture, (e-f) Normal Texture, (g-h) High Texture. Experimentally (a) was found to be too low texture to provide reliable registration (Lieberknecht et al. 2009)

Bump target (the yellow speed bump sign) due to the extremely low texture, and this was not included in the experiments. This is further discussed in Section 7.1.5.

7.1.5 Evaluation Data

To allow for consistent and repeatable results in the evaluations, an evaluation data set was created. Videos were recorded with the camera rotating around each marker for all three rotational axes. For the ground truth an inertial orientation sensor was chosen due to its high resolution and fast response time, and that it is not affected by occlusion or electro-magnetic fields. Although they are only capable of measuring rotation, these rotational components are consistent across all coordinate systems, while the scale of translational parameters varies on the marker resolution, camera calibration and camera lens properties. Visual inspection of a bounding box overlaid on the marker was used to ensure that the translational parameters were computed correctly for each experiment.

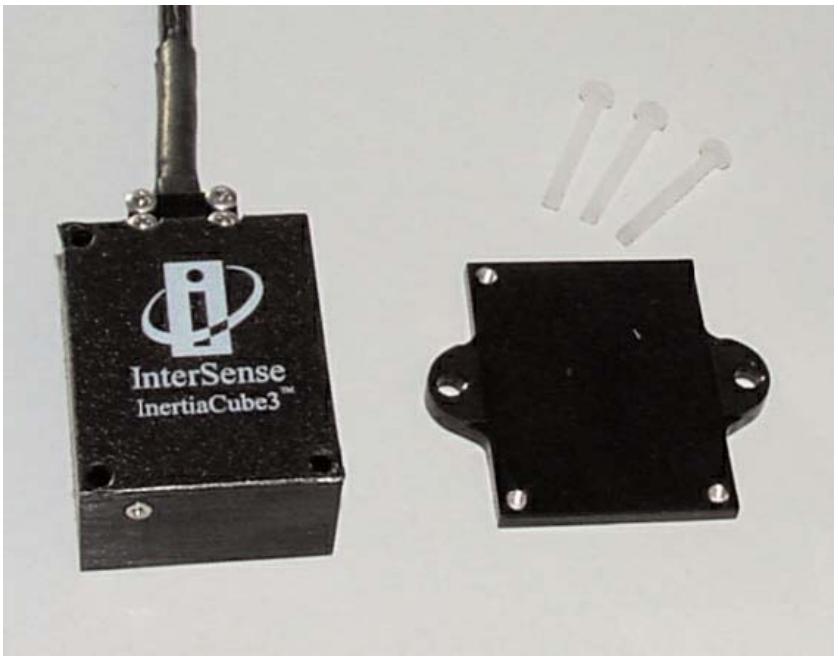


Figure 7.7: The InterSense InertiaCube3 [6]

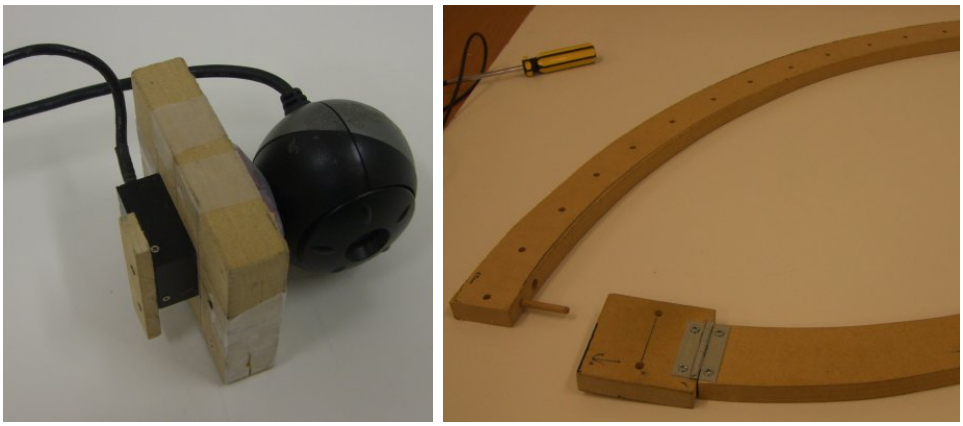
The inertial ground-truth used was the InterSense InertiaCube3[5], shown in Figure 7.7. The InertiaCube3 has an RMS accuracy of 1° yaw, 0.25° pitch and roll, and an RMS angular resolution of 0.03° [6]. The coordinate system used by the inertial orientation sensor is defined relative to the earth coordinate system; yaw determines compass orientation, pitch is the angle to the horizon, and roll the rotation around the horizon vector. This earth reference coordinate system allows the inertial sensor to be mounted in any orientation and still give the rotational value for rotation around the earth coordinate system.

A wooden mounting platform, shown in Figure 7.8(a), was constructed to securely attach the InertiaCube3 to the camera. The camera and inertial sensor were oriented to correlate the inertial tracker's pitch, yaw and roll with rotations about the x, y and z axes of the camera respectively, as shown in Figure 3.1.

[5] `http://www.intersense.com`

[6] `http://www.intersense.com/uploadedFiles/Products/IC3_datasheet_0908.pdf`

A test rig was constructed to control the rotation of the camera at a fixed distance from the marker. Figure 7.8(b) shows the wooden arms constructed to maintain a fixed rotation of the camera around the marker. The radius of the arms was 60cm, the distance at which an A4 sheet of paper completely filled the field of view of the camera vertically. The arms were attached to the base plate using a hinge to limit rotation about a single axis. For rotation around the camera's Z axis, an additional arm was made with an axle on which the mounting platform rotated.



(a) (b)

Figure 7.8: The camera and inertial orientation sensor mounting platform (a) and rotational arms (b)

To remove any effect unknown variables may have on the evaluations, registration was performed on videos captured for rotation around each axis of the marker, as shown in Figure 7.9. As each frame of video was recorded, the inertial sensors' rotational measurements were stored in a file.

All non-marker surfaces visible during the rotation of the camera were white to reduce the occurrence of erroneous feature detection, secondary lighting effects such as shadows or reflections, and any changes in camera exposure. The rotation sequences were recorded in a standard office environment, with fluorescent tube lighting of 320 lux.

Before evaluations, the ground truth coordinate system was calibrated to correlate to the natural feature registration coordinate system. This process

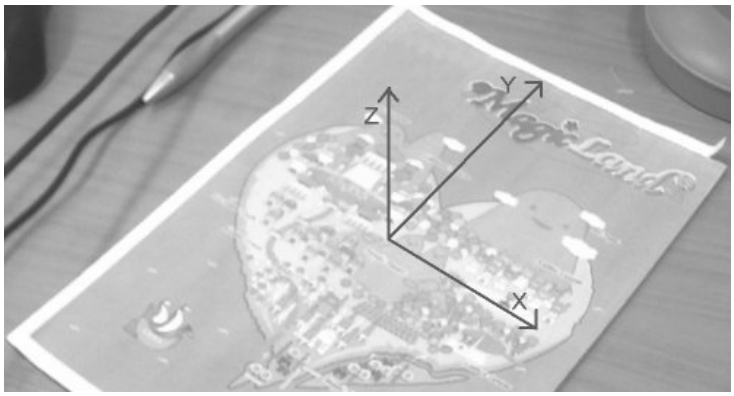


Figure 7.9: The coordinate system of the marker

is described in the following sections.

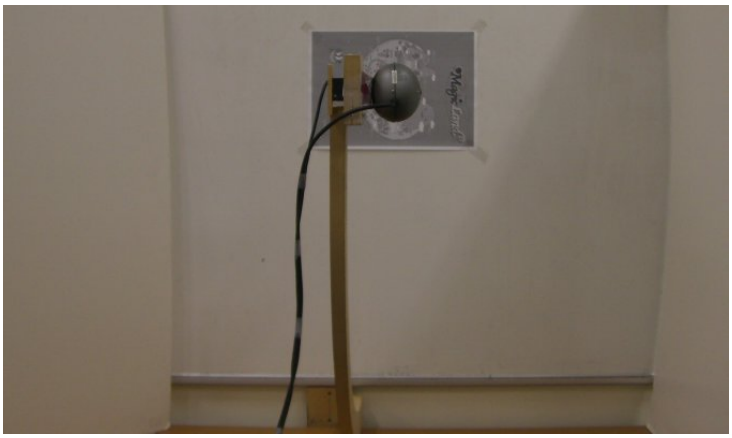
X Axis Calibration

During the X axis rotation sequence, the camera was rotated clockwise from perpendicular to the marker to the maximum rotation possible of $-81°$, looking down at the marker. From this position the camera was then rotated anticlockwise to $89°$, looking up at the marker. The camera was finally returned to perpendicular to the marker. Figure 7.10 shows the motion of the camera rig during this rotation sequence.

Frames from the X axis rotation sequence for the MagicLand marker are shown in Figure 7.11. Figure 7.11(a) shows the first captured frame, with the camera perpendicular to the marker, (b) is at approximately $-40°$ rotation, and (c) is at approximately $-81°$ rotation. Figure 7.11(d) is when the camera is at approximately $40°$ rotation, and (e) is approaching $89°$ rotation.

The captured rotation sequences were registered with the standard, optical flow and OPIRA implementations of the SIFT, SURF and Ferns registration algorithms. Figures 7.12 and 7.13 show the comparisons of the camera's X rotation when calculated using registration and when measured using the ground truth for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration.

Visual inspection of the results shows a strong trend between registration results and the ground truth, although the curves diverge over time. Because



(a)

(b)

(c)

Figure 7.10: Rig motion during the X axis rotation calibration

(a) Frame 000: $\sim 0°$ (b) Frame 030: $\sim -40°$ (c) Frame 057: $\sim -81°$

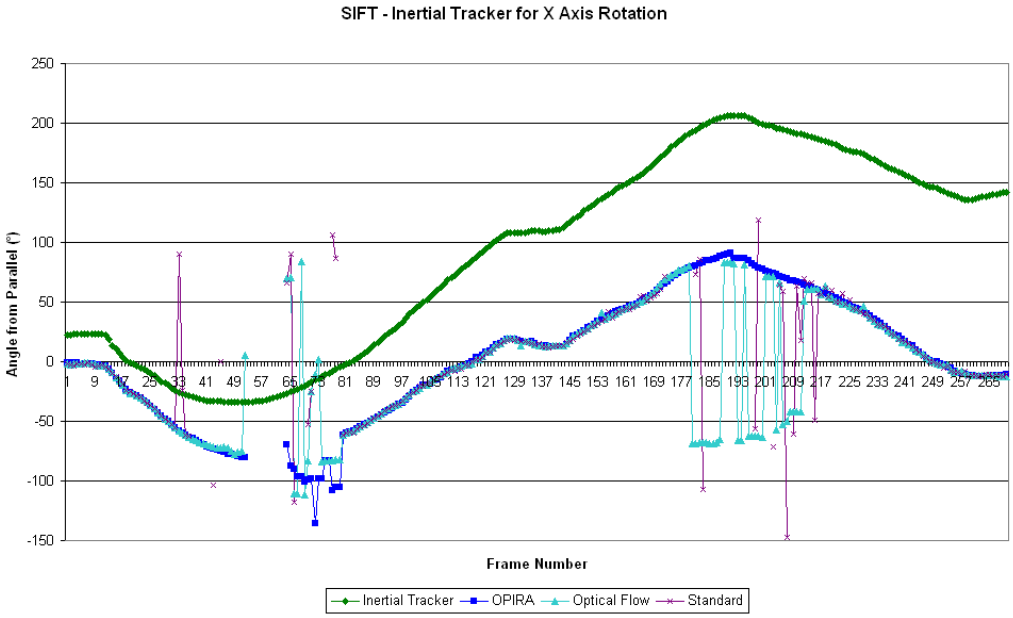
(d) Frame 160: $\sim 40°$ (e) Frame 192: $\sim 89°$

Figure 7.11: Frames from the MagicLand captured video sequence during the X axis rotation calibration

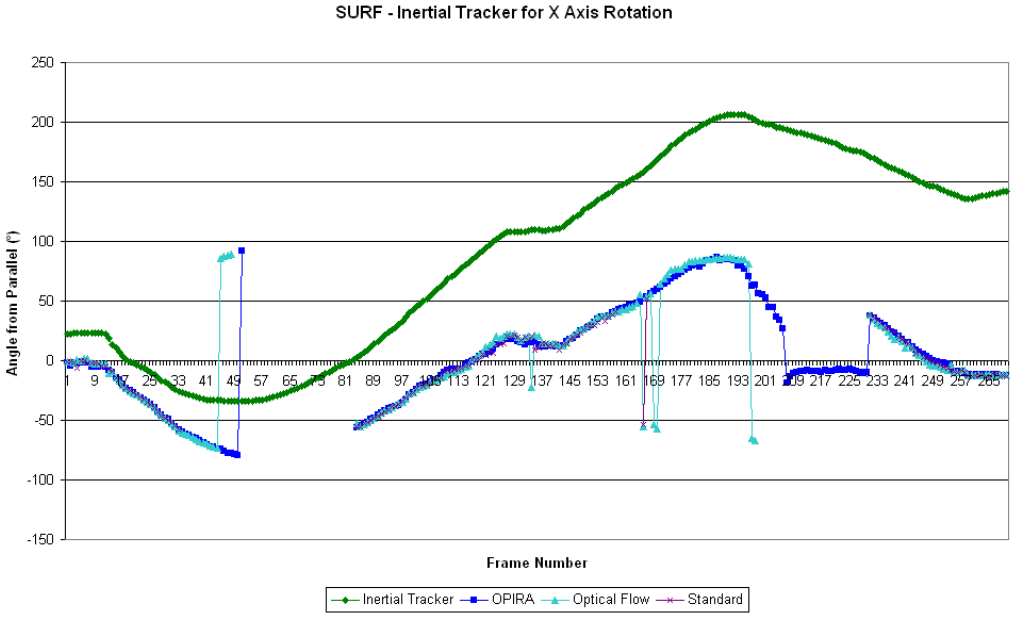
of this divergence, Mean Absolute Error (MAE) of each data point is not a useful measure of similarity. Instead, the MAE of the gradient of the line between each set of consecutive points was used, as shown in Equation 7.3, where i_x is the measured rotation of the inertial tracker at frame x, and r_x is the measured rotation of the registration at frame x.

$$MAE_{gradient} = \frac{1}{n}\sum_{x=1}^{n-1}|(i_x - i_{x+1}) - (r_x - r_{x+1})| \tag{7.3}$$

Table 7.1 shows the MAE of gradient of the registration algorithms during the X axis rotation sequence for the MagicLand marker. The minimum error is 1.38 from the OPIRA implementation of SIFT. To ensure that this is a robust measure of accuracy, the percentage of successfully registered frames was measured, and is shown in Table 7.2. The registration algorithm with the maximum percentage of successfully registered frames was the OPIRA implementation of SIFT by a significant margin, with a total of 93% of all frames within 5° of the ground truth.

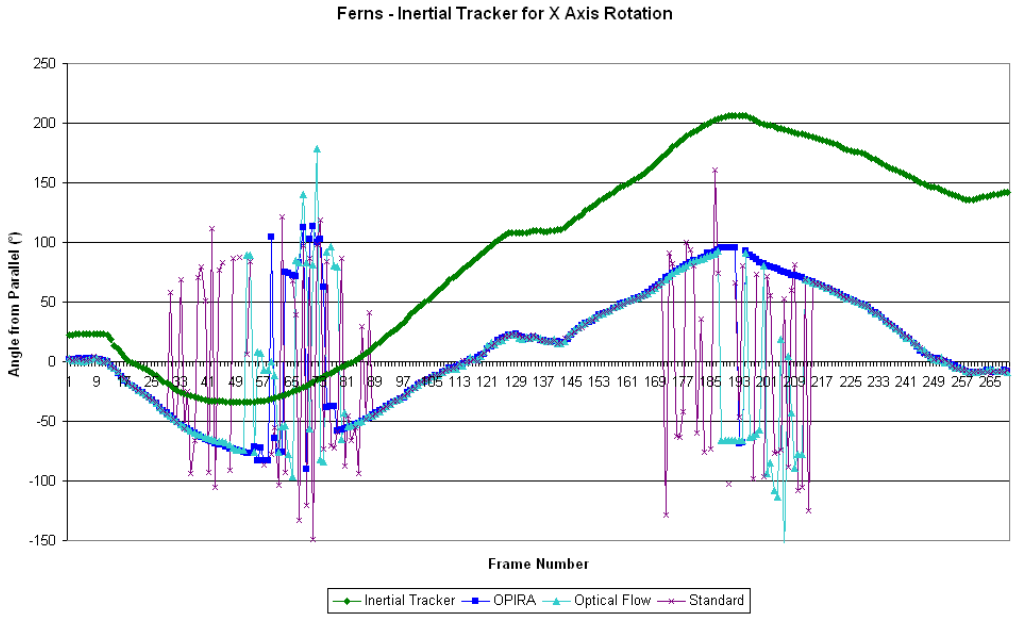


(a) SIFT



(b) SURF

Figure 7.12: SIFT and SURF registration results compared with the inertial orientation sensor ground truth for calibration of the X axis rotation sequence for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration



(a) Ferns

Figure 7.13: The Ferns registration results compared with the inertial orientation sensor ground truth for calibration of the X axis rotation sequence for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration

The OPIRA implementation of SIFT had the lowest MAE of gradient and highest percentage of successfully registered frames, making it the optimal algorithm for calibration of the ground truth for the X axis rotation sequence for the MagicLand marker.

To ensure a good calibration, all registration results outside the 5° threshold were removed. The resulting rotation values for the MagicLand marker are shown in blue in Figure 7.14, with the inertial orientation sensor ground truth values shown in dark green. Breaks in the blue line occur when registration failed when reaching the maximum rotation.

The difference between the inertial ground truth and the registration are shown by the yellow curve. The gradient of this line indicates the divergence of the two measures, due to drift in the inertial measurements. Unlike roll and pitch, gravity is not a reference in yaw calculation, and dead reckoning

	Standard	Optical flow	OPIRA
SIFT	9.44	9.59	1.38
SURF	2.29	5.43	2.18
Ferns	34.94	13.59	6.28

Table 7.1: Mean Average Error of Gradient for rotation about X axis for the MagicLand Marker

	Standard	Optical flow	OPIRA
SIFT	63%	86%	93%
SURF	52%	69%	84%
Ferns	64%	87%	93%

Table 7.2: Percentage of successfully registered frames for rotation about X axis for the MagicLand Marker

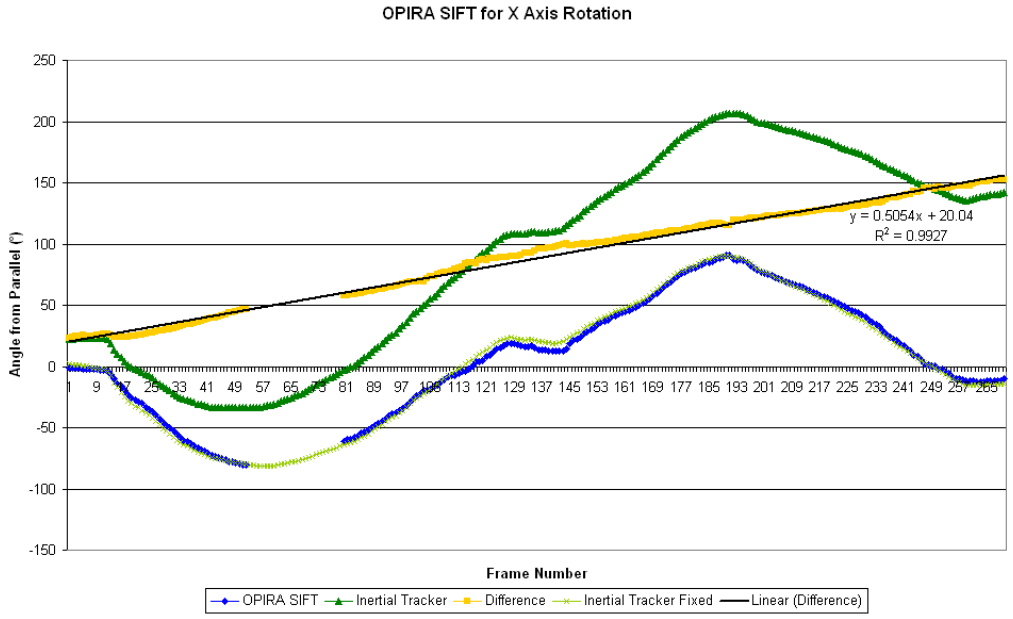


Figure 7.14: The ground truth data fitted to the camera calibration data for rotation about the X axis for the MagicLand marker. Breaks in the blue line are where registration failed or error was greater than 5°

must be used. The drift in the inertial orientation sensor results is due to compounding errors during the dead reckoning calculations.

The line of best fit was found for the difference values to calculate the drift, which was approximately 0.5° per frame. The Y intercept of 20.04° is due to a combination of drift occurring between initialisation of the inertial orientation sensor and the start of recording, and slight misalignment of the sensor and camera in the mounting rig. The very high R^2 value of 0.9927 confirms the relationship between the difference and time, proving that the divergence is due to drift.

Using the formula for the line of best fit, the ground truth coordinate system was calibrated to match the registration coordinate system, shown in Figure 7.14 as the light green curve.

This calibration process was carried out for each marker's video sequences. For all videos, the OPIRA implementation of SIFT provided the highest correlation between the ground truth and camera registration. The MAE for calibration of each marker is shown in Table 7.3 and the percentage of successfully registered frames for calibration of each marker is shown in Table 7.4.

Marker	MAE
MagicLand	2.92°
Bump	6.08°
Stop	2.38°
Lucent	1.53°
MacMini	1.47°
Isetta	1.59°
Philadelphia	1.25°
Grass	1.86°
Wall	1.41°

Table 7.3: The MAE of X axis calibration for all markers

All the markers had a MAE of less than 3° and a minimum of 65% successfully registered frames with the exception of the Bump marker, which did not have enough detail for any of the registration algorithms to work well.

Marker	% Successful
MagicLand	93.1%
Bump	17.6%
Stop	67.9%
Lucent	68.9%
MacMini	78.5%
Isetta	73.4%
Philadelphia	84.1%
Grass	73.6%
Wall	77.2%

Table 7.4: Percentage of successfully registered frames of X axis calibration for all markers

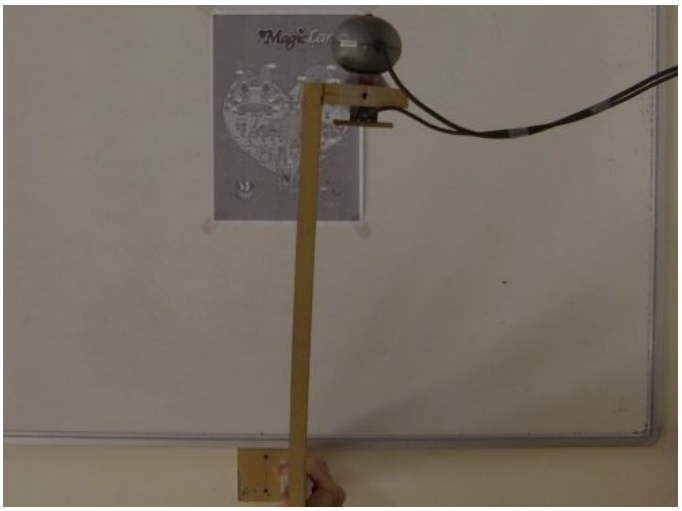
Y Axis Calibration

During the Y axis rotation sequence, the camera was rotated clockwise from perpendicular to the marker to the maximum rotation possible of 78°, on the right of the marker. From this position the camera was then rotated anticlockwise to −86°, on the left of the marker. The camera was then returned to perpendicular to the marker. Figure 7.15 shows the motion of the camera rig during this rotation sequence.

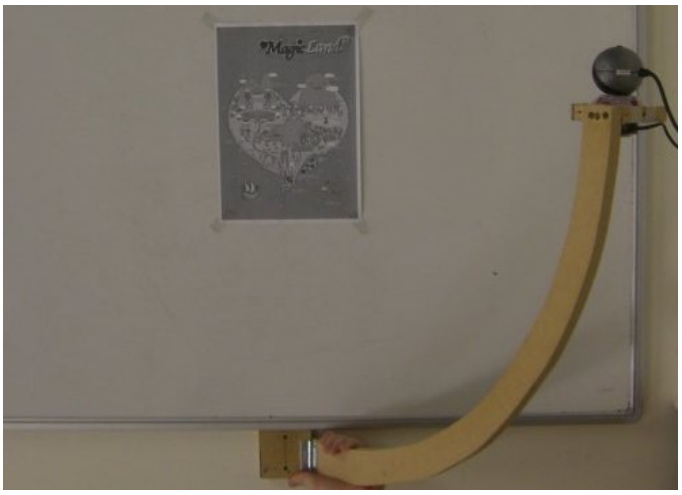
Frames from the Y axis rotation sequence for the MagicLand marker are shown in Figure 7.16. Figure 7.16(a) shows the first captured frame, with the camera perpendicular to the marker, (b) is at approximately 40° rotation, and (c) is at approximately 78° rotation. Figure 7.16(d) is when the camera is at approximately −40° rotation, and (e) is at approximately −86° rotation.

The captured rotation sequences were registered with the standard, optical flow and OPIRA implementations of the SIFT, SURF and Ferns registration algorithms. Figures 7.17 and 7.18 show the comparisons of the camera's Y rotation when calculated using registration and when measured using the ground truth for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration.

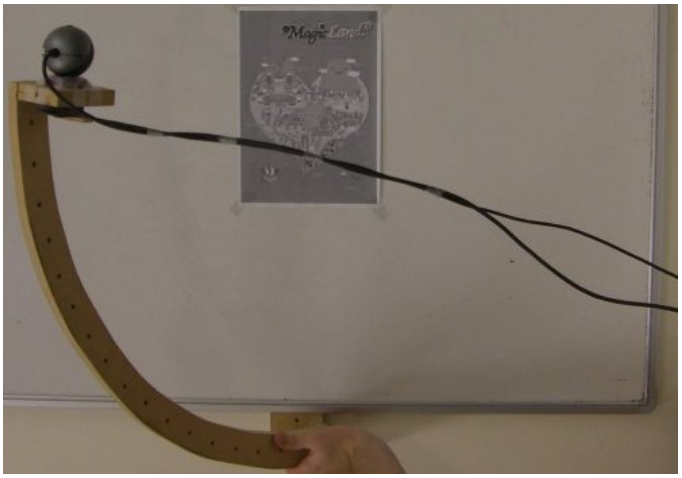
Table 7.5 shows the MAE of gradient of the registration algorithms during the Y axis rotation sequence for the MagicLand marker. Although the minimum error is 1.11 from the standard implementation of SURF, this al-



(a)



(b)



(c)

Figure 7.15: Rig motion during the Y axis rotation calibration

(a) Frame 000: $\sim 0°$ (b) Frame 025: $\sim 40°$ (c) Frame 052: $\sim 78°$

(d) Frame 115: $\sim -40°$ (e) Frame 136: $\sim -86°$

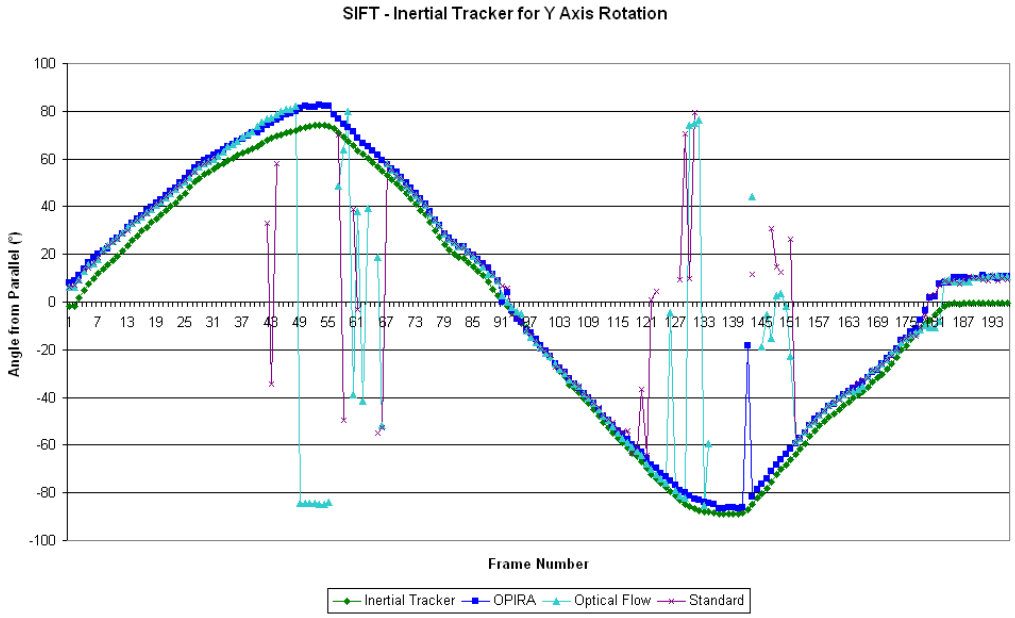
Figure 7.16: Frames from the MagicLand captured video sequence during the Y axis rotation calibration

gorithm only had a 54% success rate, as shown in Table 7.6. The OPIRA implementation of SIFT had a slightly higher error of 1.17, due to a single erroneous registration, but had the highest success rate of 98%. With this erroneous registration removed, the error of the OPIRA implementation of SIFT drops to 0.50.

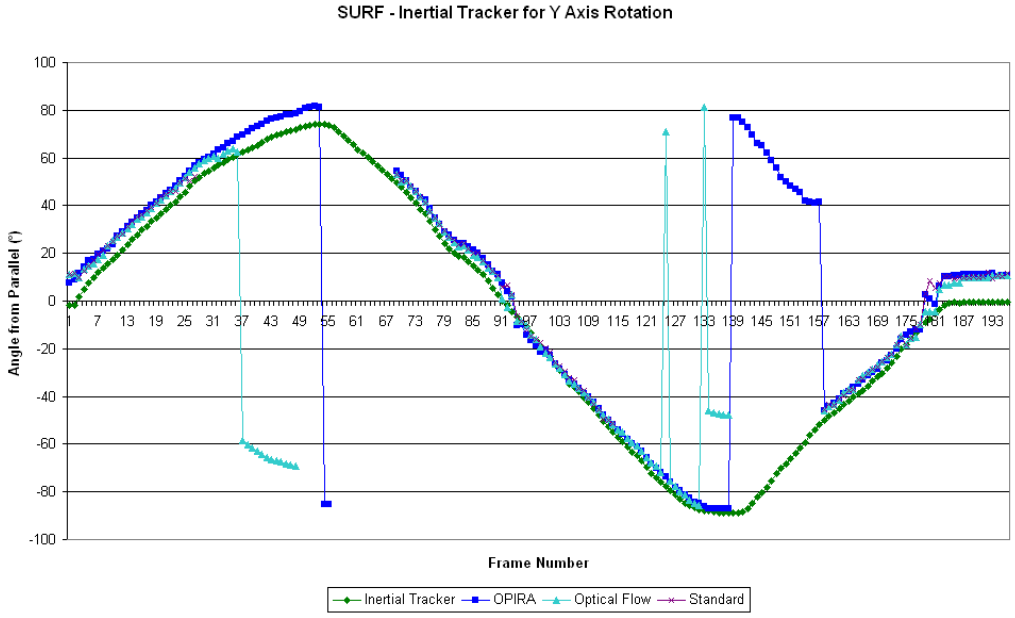
Once the single erroneous registration was removed, the OPIRA implementation of SIFT had the lowest MAE of gradient and highest percentage of successfully registered frames, making it the optimal algorithm for calibration of the ground truth for the Y axis rotation sequence for the MagicLand marker.

To ensure a good calibration, all registration results outside the 5° threshold were removed. The resulting rotation values for the MagicLand marker are shown in blue in Figure 7.19, with the inertial orientation sensor ground truth values shown in dark green.

To ensure there was no drift the difference between the two data sets was calculated, as shown in yellow. The line of best fit for the difference is shown

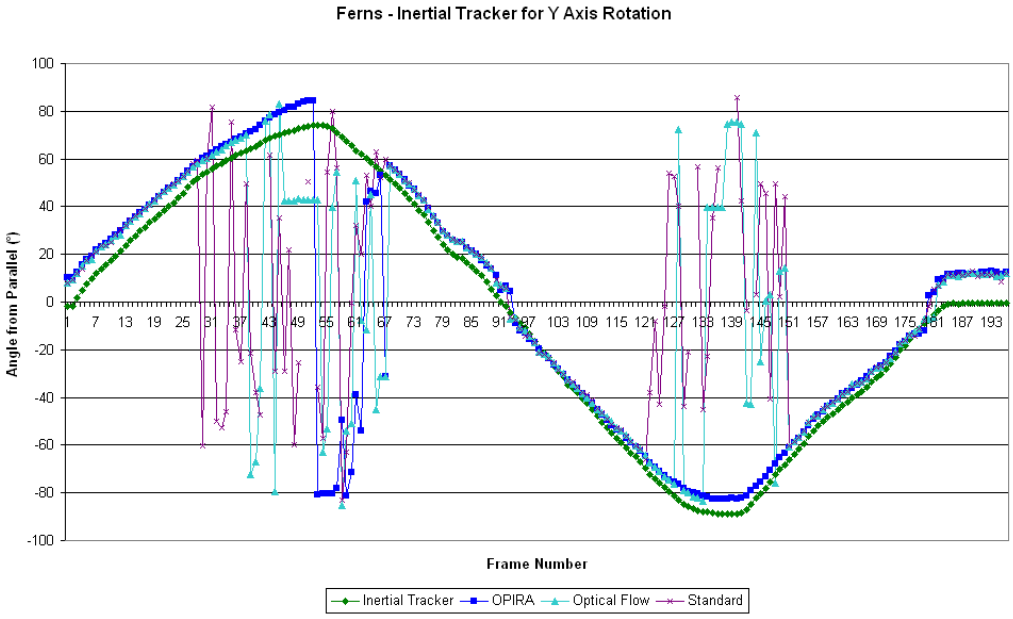


(a) SIFT



(b) SURF

Figure 7.17: SIFT and SURF registration results compared with the inertial orientation sensor ground truth for calibration of the Y axis rotation sequence for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration



(a) Ferns

Figure 7.18: Ferns registration results compared with the inertial orientation sensor ground truth for calibration of the Y axis rotation sequence for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration

	Standard	Optical flow	OPIRA
SIFT	6.88	7.99	1.17
SURF	1.11	5.48	3.26
Ferns	15.84	13.51	3.45

Table 7.5: Mean Average Error of gradient for rotation about Y axis for the MagicLand marker

	Standard	Optical flow	OPIRA
SIFT	64%	82%	98%
SURF	54%	74%	87%
Ferns	65%	84%	93%

Table 7.6: Percentage of successfully registered frames for rotation about Y axis for the MagicLand marker

in black, where the low R^2 value of 0.023 suggests no correlation between the difference and time, indicating that there was no inertial drift over time.

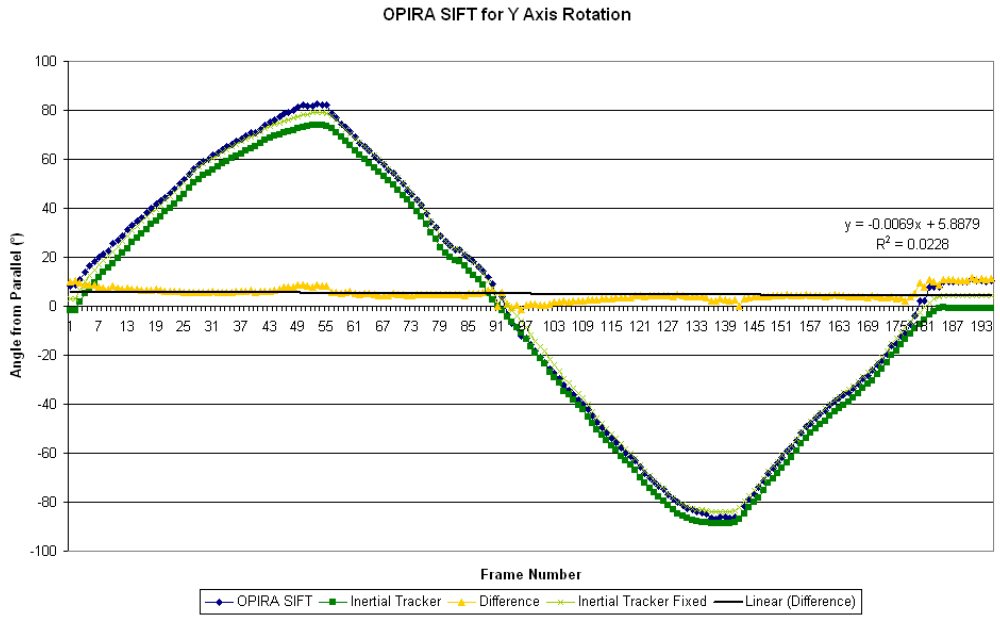


Figure 7.19: The ground truth data fitted to the camera calibration data for rotation about the Y axis for the MagicLand marker

The difference between the two data sets is due to a combination of misalignment between the sensor and camera in the mounting rig. To account for the misalignment, the average of the difference, which was found to be 4.88°, was added to the ground truth results, shown in the graph as a light green line.

This calibration process was carried out for each marker's video sequences. For all videos, the OPIRA implementation of SIFT provided the highest correlation between the ground truth and camera registration. The MAE for calibration of each marker is shown in Table 7.7 and the percentage of successfully registered frames for calibration of each marker is shown in Table 7.8.

All the markers had a MAE of less than 4° and a minimum of 70% successfully registered frames with the exception of the Bump marker, which

Marker	MAE
MagicLand	1.93°
Bump	3.36°
Stop	2.49°
Lucent	3.78°
MacMini	1.45°
Isetta	3.00°
Philadelphia	2.43°
Grass	2.89°
Wall	1.61°

Table 7.7: The MAE of Y axis calibration for all markers

Marker	% Successful
MagicLand	98.0%
Bump	28.3%
Stop	73.7%
Lucent	78.3%
MacMini	70.0%
Isetta	75.3%
Philadelphia	77.8%
Grass	71.7%
Wall	88.4%

Table 7.8: Percentage of successfully registered frames of Y axis calibration for all markers

did not have enough detail for any of the registration algorithms to work well.

Z Axis Calibration

During the Z axis rotation sequence, the camera was rotated clockwise from vertically aligned to the marker to approximately 180° rotation. From this position the camera was then rotated anticlockwise to approximately −180° rotation. The camera was then returned to vertical alignment to the marker. Figure 7.20 shows the motion of the camera rig during this rotation sequence.

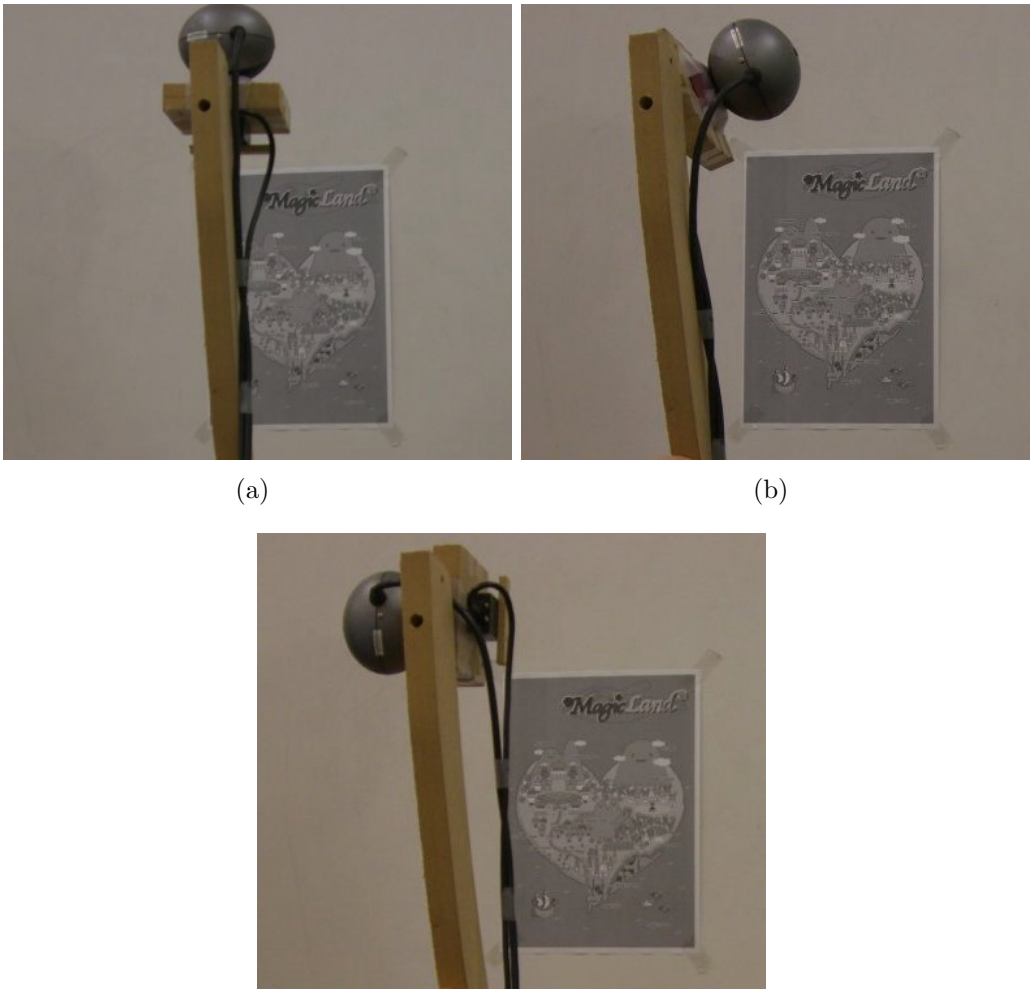
Frames from the Z axis rotation sequence for the MagicLand marker are shown in Figure 7.21. Figure 7.21(a) shows the first captured frame, with the camera vertically aligned to the marker, (b) is at approximately 90° rotation, and (c) is at approximately 180° rotation. Figure 7.16(d) is when the camera is at approximately −90° rotation, and (e) is at approximately −180° rotation.

The centre of rotation of the camera during the Z axis rotation sequences was not the centre of the markers, as the axis of rotation was offset from the camera's centre. Despite this, rotation was still isolated to the Z axis, and the majority of each marker was still in frame during the rotation sequence.

The captured rotation sequence was registered with the standard, optical flow and OPIRA implementations of the SIFT, SURF and Ferns registration algorithms. Figures 7.22 and 7.23 show the comparisons of the camera's Z rotation when calculated using registration and when measured using the ground truth for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration.

Table 7.9 shows the MAE of gradient of the registration algorithms during the Z axis rotation sequence for the MagicLand marker. The minimum error is 0.53 for the OPIRA implementation of SIFT. Table 7.10 shows the percentage of successfully registered frames, for this sequence the OPIRA implementations of SIFT, SURF and all implementations of Ferns had a 100% success rate.

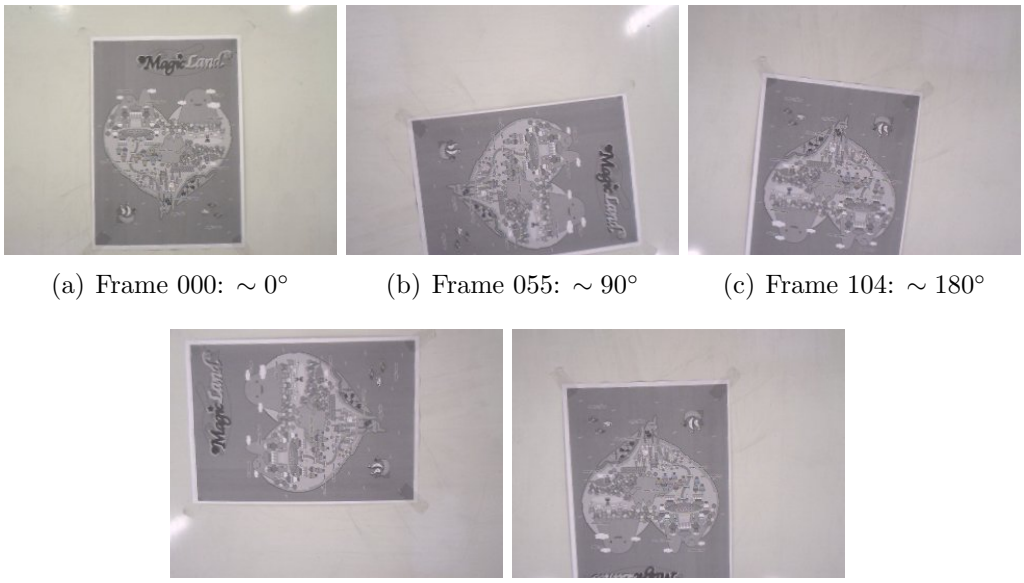
The OPIRA implementation of SIFT had the lowest MAE of gradient and highest equal percentage of successfully registered frames, making it the



(a) (b)

(c)

Figure 7.20: Rig motion during the Z axis rotation calibration



(a) Frame 000: $\sim 0°$ (b) Frame 055: $\sim 90°$ (c) Frame 104: $\sim 180°$

(d) Frame 191: $\sim -90°$ (e) Frame 232: $\sim -180°$

Figure 7.21: Frames from the MagicLand captured video sequence during the Z axis rotation calibration

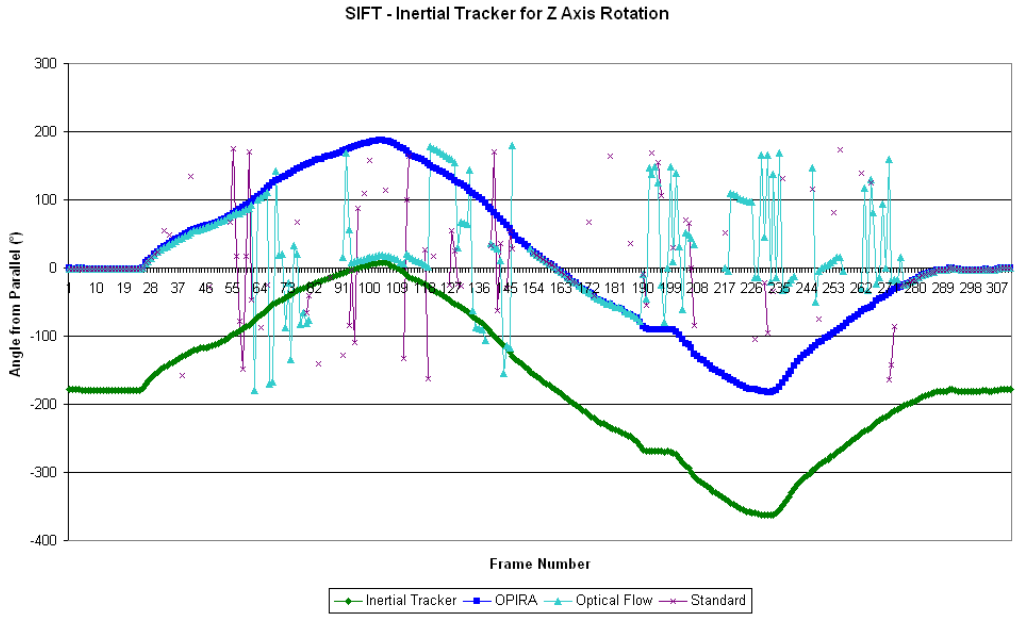
optimal algorithm for calibration of the ground truth for the Z axis rotation sequence for the MagicLand marker.

No registration results were needed to be removed for the Z axis rotation MagicLand sequence calibration. The rotation values are shown in blue in Figure 7.24, with the inertial orientation sensor ground truth values shown in dark green.

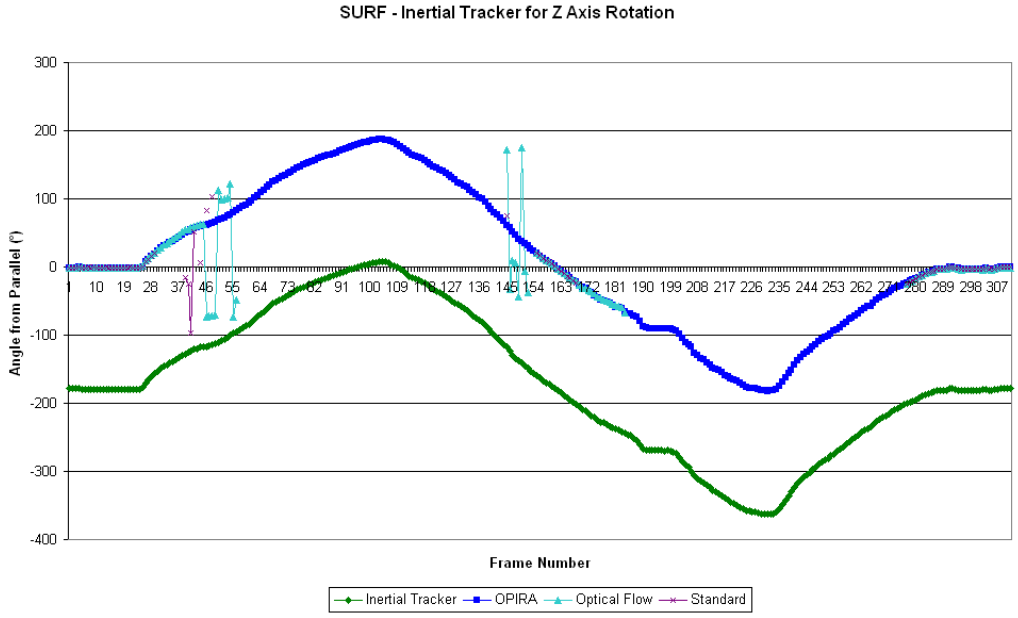
The correlation between the two lines is apparent, however there is a separation of approximately 180°, due to the orientation of the inertial orientation sensor coordinate system to the camera coordinate system.

To ensure there was no drift the difference between the two data sets was calculated, shown in yellow. The line of best fit for the difference is shown in black, where the low R^2 value of 0.02 suggests that there is no correlation between the difference and time, indicating that there was no inertial drift over time.

The difference between the two data sets is due to misalignment between the sensor and camera in the mounting rig. To account for the misalignment,

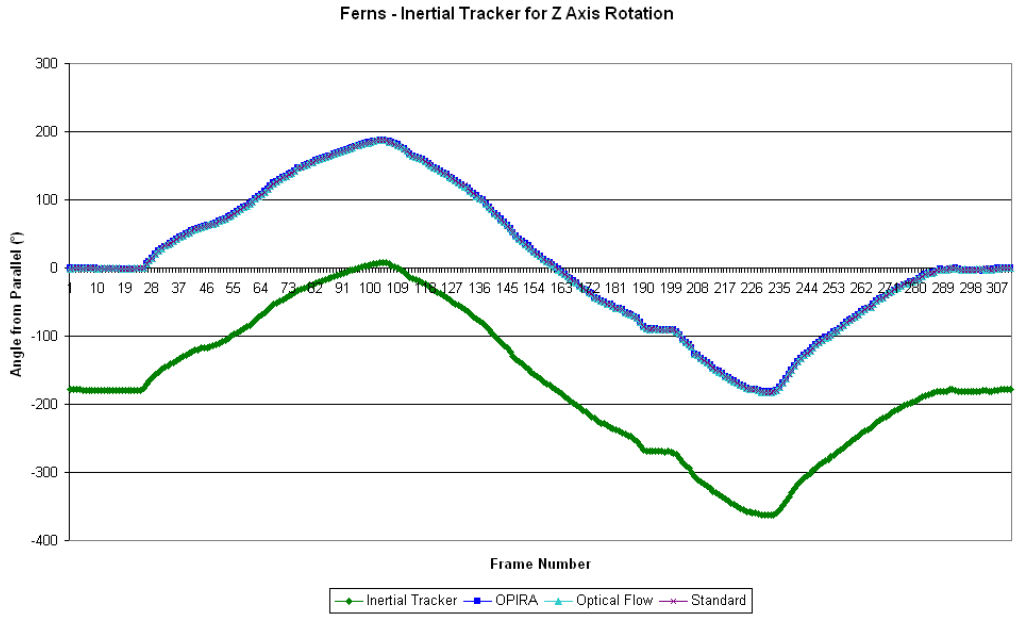


(a) SIFT



(b) SURF

Figure 7.22: SIFT and SURF registration results compared with the inertial orientation sensor ground truth for calibration of the Z axis rotation sequence for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration



(a) Ferns

Figure 7.23: Ferns registration results compared with the inertial orientation sensor ground truth for calibration of the Z axis rotation sequence for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration

the average of the difference, which was found to be $-178.93°$, was added to the ground truth results, shown in the graph as a light green line.

This calibration process was carried out for each marker's video sequences. For all videos, the OPIRA implementation of SIFT provided the highest correlation between the ground truth and camera registration. The MAE for calibration of each marker is shown in Table 7.11 and the percentage of successfully registered frames for calibration of each marker is shown in Table 7.12.

All the markers had a MAE of less than 3° and 100% successfully registered frames with the exception of the Bump marker, which did not have enough detail for any of the registration algorithms to work well.

The results of ground truth calibration for each marker's three rotational video sequences were accurate, with the exception of the Bump marker, which

	Standard	Optical flow	OPIRA
SIFT	27.50	29.53	0.53
SURF	3.56	10.93	0.56
Ferns	0.68	0.65	0.60

Table 7.9: Mean Average Error of gradient for rotation about Z axis for the MagicLand marker

	Standard	Optical flow	OPIRA
SIFT	26%	65%	100%
SURF	24%	36%	100%
Ferns	100%	100%	100%

Table 7.10: Percentage of successfully registered frames for rotation about Z axis for the MagicLand marker

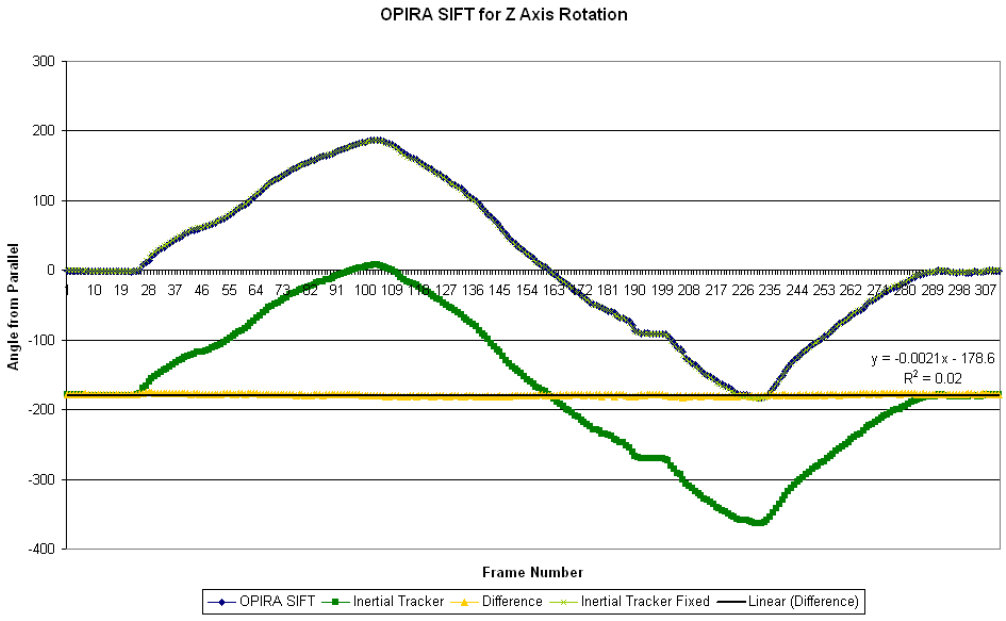


Figure 7.24: The ground truth data fitted to the camera calibration data for rotation about the Z axis for the MagicLand marker

Marker	MAE
MagicLand	1.12°
Bump	7.78°
Stop	2.08°
Lucent	2.01°
MacMini	2.22°
Isetta	2.00°
Philadelphia	1.93°
Grass	2.08°
Wall	1.84°

Table 7.11: The MAE of Z axis calibration for all markers

Marker	% Successful
MagicLand	100.0%
Bump	21.9%
Stop	100.0%
Lucent	100.0%
MacMini	100.0%
Isetta	100.0%
Philadelphia	100.0%
Grass	100.0%
Wall	100.0%

Table 7.12: Percentage of successfully registered frames of Z axis calibration for all markers

had consistently poor accuracy regardless of rotation sequence or registration algorithm. As these calibration videos represented the optimal registration conditions, registration after the image transformations and deformations discussed in the previous chapters would only reduce the accuracy of registration. For this reason, the Bump marker was not included in any of the evaluations.

7.2 OPIRA

In Chapter 5 a new algorithm called Optical-flow Perspective Invariant Registration Augmentation (OPIRA) was discussed, which combines optical flow

with image rectification to improve the invariance of natural feature algorithms to changes in scale, rotation, and perspective.

In Section 7.2.1 prior work involving a preliminary evaluation of OPIRA is presented. This evaluation is based on visual inspection, and investigates how OPIRA improves user experience in certain applications such as Augmented Reality.

Section 7.2.2 describes a quantitative evaluation of the perspective invariance which can be achieved with OPIRA using the inertial ground truth described previously, and in Section 7.2.3 this ground truth is used to perform a quantitative evaluation of the rotation invariance which can be achieved with OPIRA.

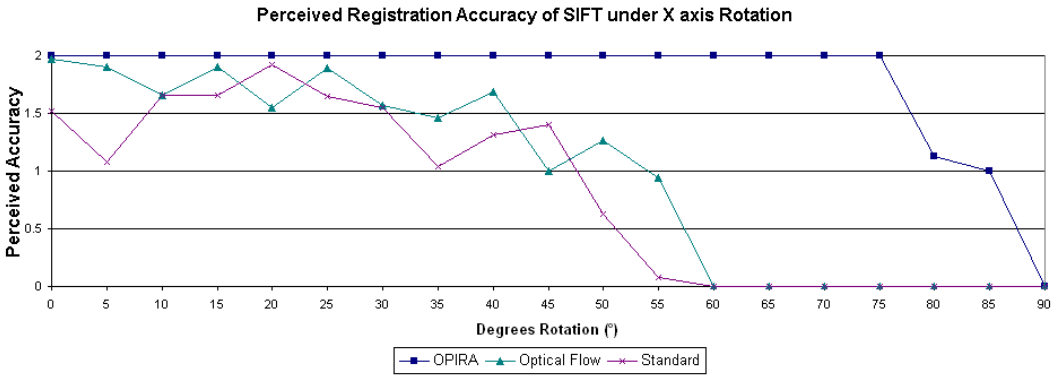
Section 7.2.4 examines the efficiency of the internal OPIRA selection algorithm, for the purposes of improvements to operating speed such as Fast-OPIRA.

7.2.1 Visual Inspection

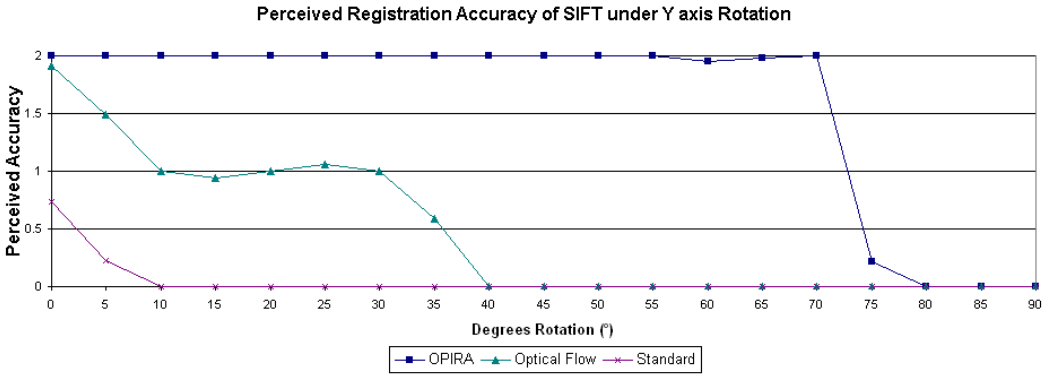
A preliminary evaluation of the visual improvement in the accuracy of registration using OPIRA was conducted by Clark et al. (2008) using the rotation dependent SIFT and SURF registration algorithms. A video sequence was captured with camera being rotated 90° around each axis of the MagicLand marker at a distance of 50cm. The accuracy of registration was evaluated by visually inspecting the alignment of a virtual rectangle rendered on the marker. Each registration was assigned a value from 0 to 2; 0 indicating a serious misaligned registration or registration failure, 1 indicating a slight misalignment, and 2 indicating a visually perfect registration.

The average accuracy value was calculated every 5° for the OPIRA, optical flow and standard implementations, and the results are shown in Figures 7.25 and 7.26.

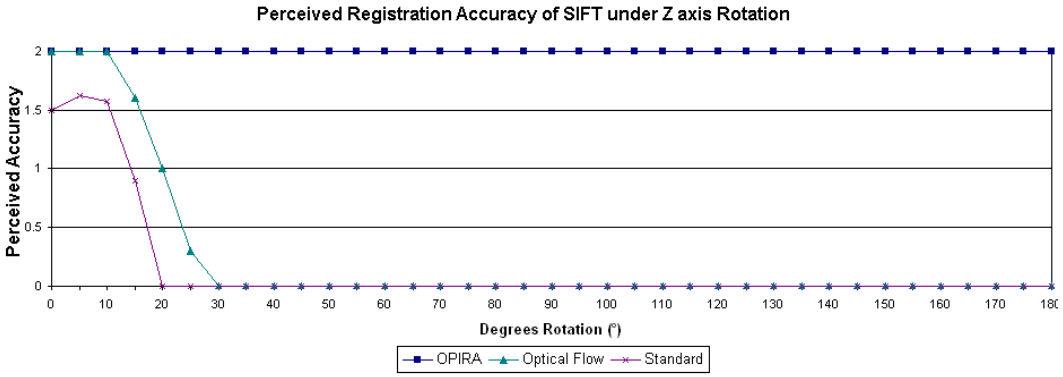
The SIFT algorithm visually improved as the camera approached parallel to the marker using the OPIRA implementation. In the X axis rotation sequence the point where the registration began to degrade increased from 25° to 75°, with the amount of rotation until failure increasing from 60° to almost 90°. In the Y axis rotation sequence the point where the registration



(a) X axis Rotation

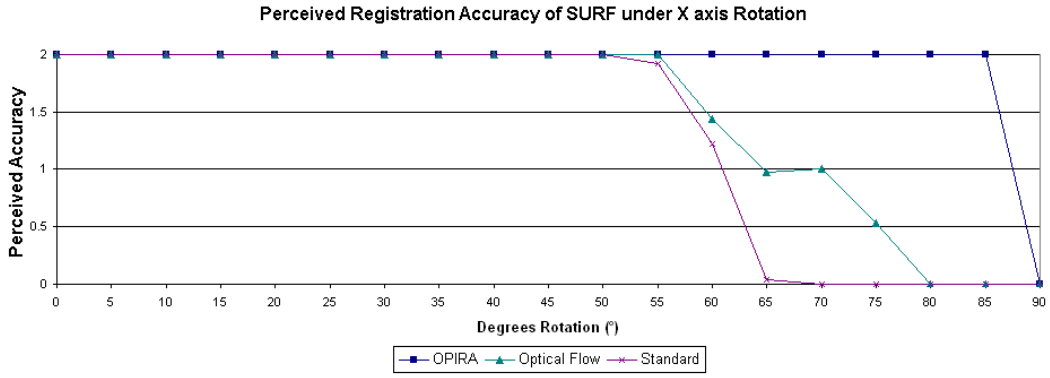


(b) Y axis Rotation

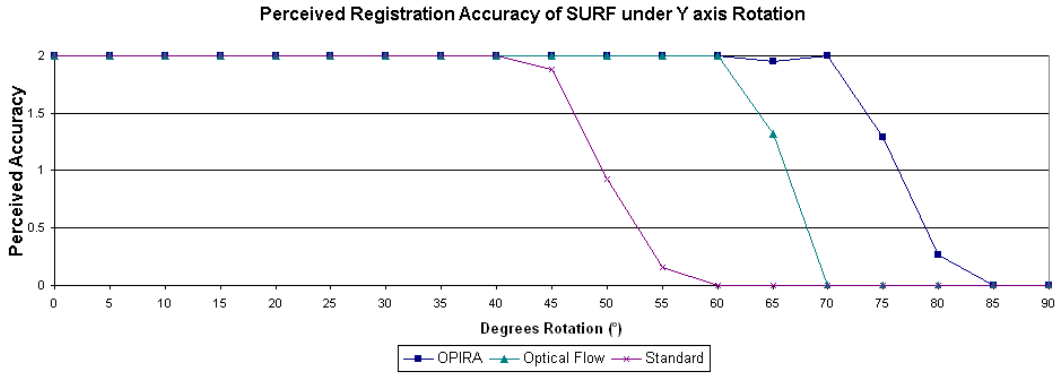


(c) Z axis Rotation

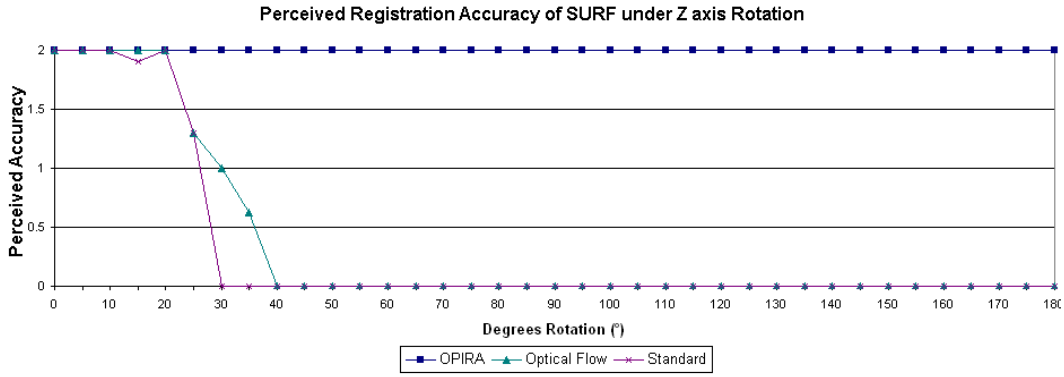
Figure 7.25: Perceived registration accuracy of the standard, optical flow and OPIRA implementations of the SIFT algorithm. 0 indicates serious misalignment, 1 indicates slight misalignment, 2 is perfectly aligned visually (Clark et al. 2008)



(a) X axis Rotation



(b) Y axis Rotation



(c) Z axis Rotation

Figure 7.26: Perceived registration accuracy of the standard, optical flow and OPIRA implementations of the SIFT algorithm. 0 indicates serious misalignment, 1 indicates slight misalignment, 2 is perfectly aligned visually (Clark et al. 2008)

began to degrade increased from 0° to 70°, with the amount of rotation until failure increasing from 10° to 80°. In the Z axis rotation sequence the rotation dependent implementation of SIFT began to degrade after just 10° and failed at 20°, while the OPIRA implementation achieved full 360° rotation invariance.

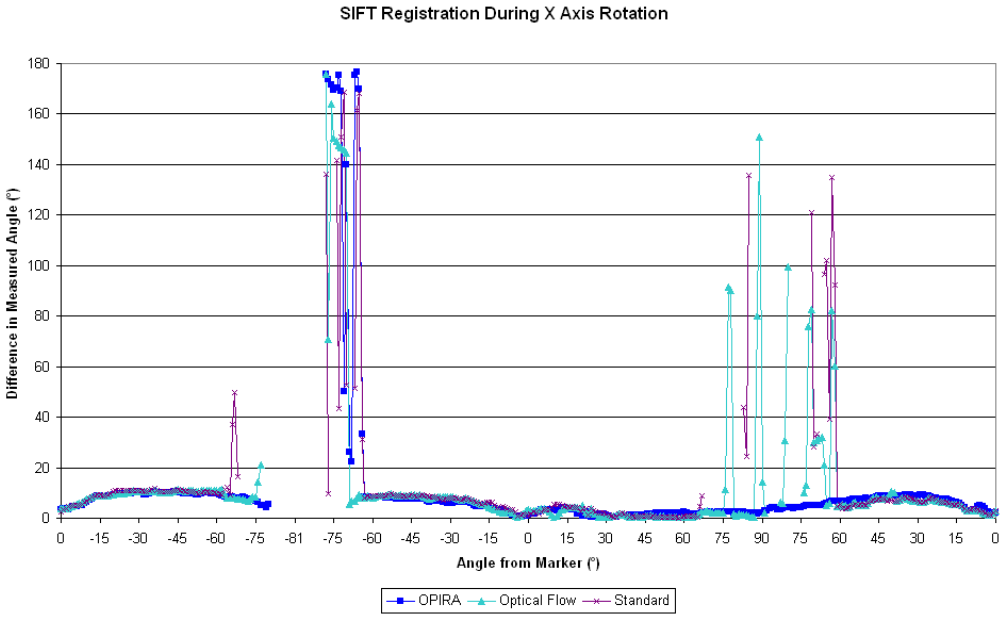
Similarly, the SURF algorithm visually improved as the camera approached parallel to the marker using the OPIRA implementation. In the X axis rotation sequence the point where the registration began to degrade increased from 55° to 85°, with the amount of rotation until failure increasing from 70° to almost 90°. In the Y axis rotation sequence the point where the registration began to degrade increased from 40° to 70°, with the amount of rotation until failure increasing from 30° to 85°. In the Z axis rotation sequence the rotation dependent implementation of SURF began to degrade after just 20° and failed at 30°, while the OPIRA implementation achieved full 360° rotation invariance.

7.2.2 Perspective Invariance

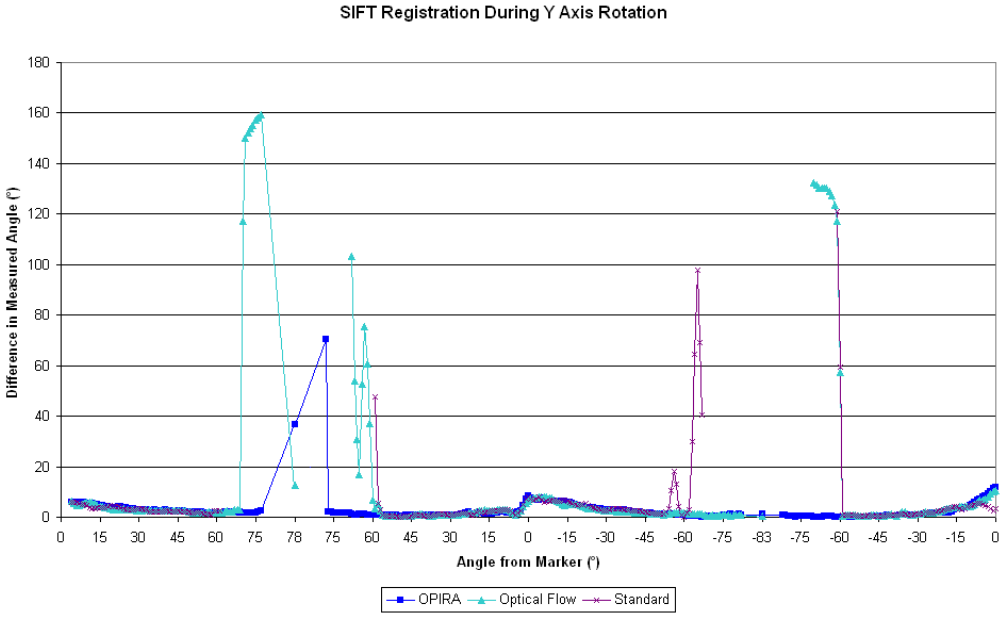
The preliminary results obtained in the Section 7.2.1 show a large visual improvement in the accuracy of registration when using the OPIRA registration implementation compared to the standard and optical flow implementations as the marker undergoes perspective distortion. In this section the improvements possible using OPIRA implementations of natural feature registration algorithms when the marker is distorted due to changes in perspective are empirically evaluated using the inertial orientation sensor ground truth image sequences captured in Section 7.1.5.

For clarity, the graphs in this section only show the difference between the rotation angle calculated using registration and measured using the ground truth. This difference is compared to the angle in rotation from perpendicular to the marker. As discussed in Section 7.1.5, the physical limits of the testing rig were −81° and 90° for the X axis, and −83° and 78° for the Y axis.

SIFT Figure 7.27 shows the differences between the rotation angle calculated using the three SIFT implementations and measured using the ground truth for the MagicLand marker. During rotation around the X axis, the



(a) X axis Rotation



(b) Y axis Rotation

Figure 7.27: Difference in measured angle for the standard, optical flow and OPIRA implementations of SIFT for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration

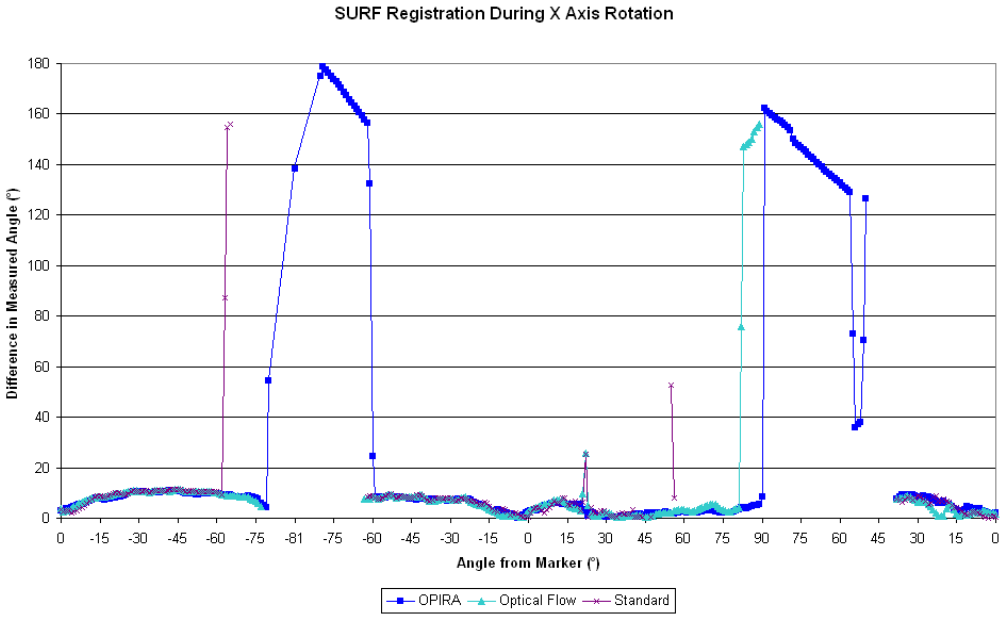
standard registration implementation, shown in purple, fails to register after -67° rotation in the negative direction and 67° rotation in the positive direction. The optical flow implementation, shown in light blue, fails to register after -77° rotation in the negative direction but succeeds for all rotations in the positive direction, although there are several frames with high error. The OPIRA implementation, shown in dark blue, fails at -80° rotation in the negative direction but successfully registers all frames with low error during rotation in the positive direction.

In the Y axis rotation sequence the standard implementation fails after 62° rotation in the positive direction and -62° rotation in the negative direction. The optical flow implementation only fails briefly during the negative and positive rotations, although the difference between the ground truth value and registration angle is large. The OPIRA implementation succeeds registration on all frames, although there is one poorly registered frame during positive rotation.

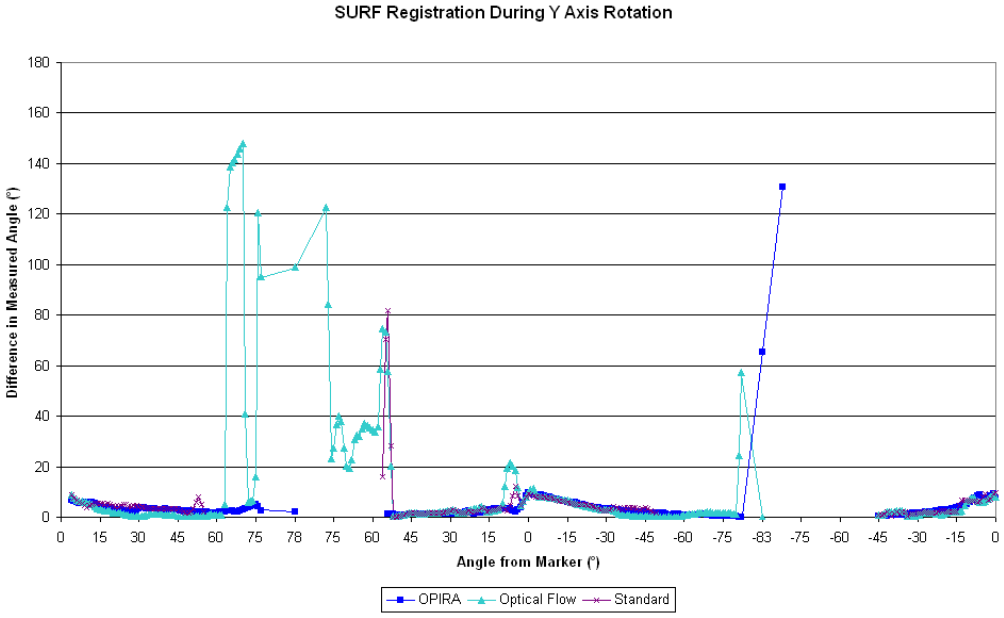
SURF Figure 7.28 shows the differences between the rotation angle calculated using the three SURF implementations and measured using the ground truth for the MagicLand marker. During rotation around the X axis, the standard registration implementation, shown in purple, fails to register after -62° rotation in the negative direction and 50° rotation in the positive direction. The optical flow implementation, shown in light blue, fails to register after -77° rotation in the negative direction and 81° rotation in the positive direction. The OPIRA implementation, shown in dark blue, fails at -79° rotation in the negative direction and 90° rotation in the positive direction.

In the Y axis rotation sequence the standard implementation fails after 55° rotation in the positive direction and -49° rotation in the negative direction. The optical flow implementation fails at 62° rotation, and −80° rotation. The OPIRA implementation fails after 78° rotation and again at −83° rotation.

Ferns Figure 7.29 shows the differences between the rotation angle calculated using the three Ferns implementations and measured using the ground truth for the MagicLand marker. During rotation around the X axis, the

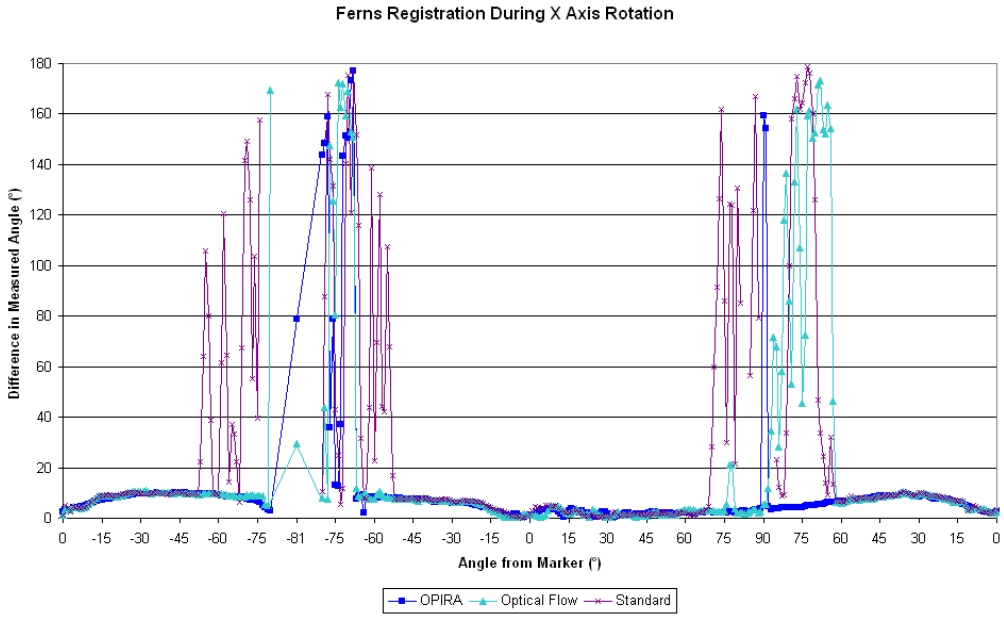


(a) X axis Rotation

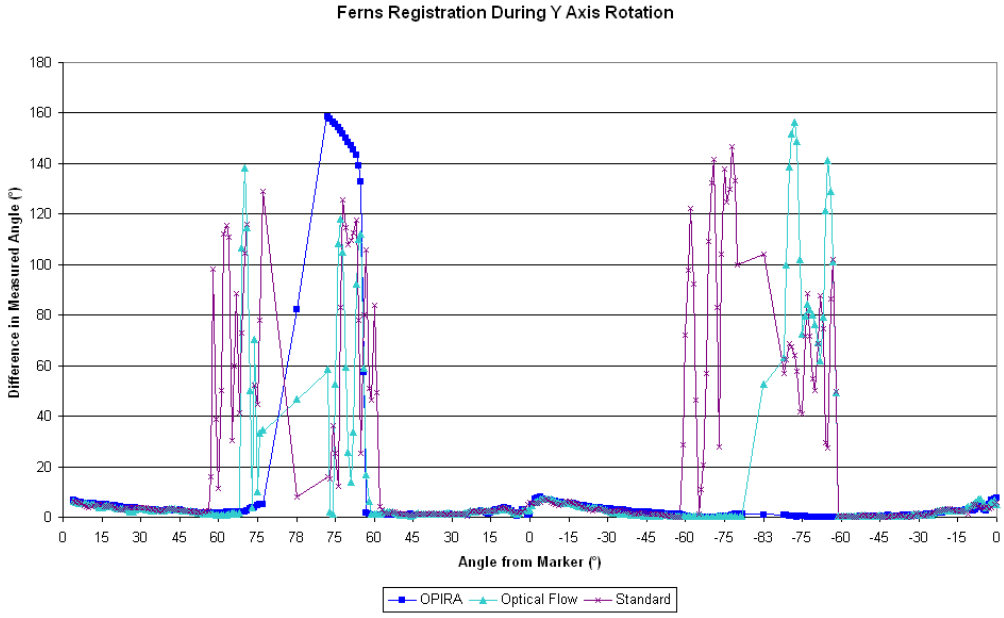


(b) Y axis Rotation

Figure 7.28: Difference in measured angle for the standard, optical flow and OPIRA implementations of SURF for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration



(a) X axis Rotation



(b) Y axis Rotation

Figure 7.29: Difference in measured angle for the standard, optical flow and OPIRA implementations of the Ferns classifier for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration

standard registration implementation, shown in purple, fails to register after -52° rotation in the negative direction and 69° rotation in the positive direction. The optical flow implementation, shown in light blue, fails to register after -79° rotation in the negative direction and 88° rotation in the positive direction. The OPIRA implementation, shown in dark blue, fails at -81° rotation in the negative direction and 90° rotation in the positive direction.

In the Y axis rotation sequence the standard implementation fails after 56° rotation in the positive direction and -58° rotation in the negative direction. The optical flow implementation fails at 68° rotation, and −83° rotation. The OPIRA implementation fails after 78° rotation, however registration succeeds for all rotations in the negative direction.

Overall Results Tables 7.13, 7.14 and 7.15 show the mean absolute error of the measured angle, average number of feature matches, and percent of successfully registered frames for the standard, optical flow and OPIRA implementations of the SIFT registration algorithm. For almost all markers, the OPIRA implementation had a lower mean error than both the optical flow and standard implementations of the SIFT algorithm. The only exception was for the X rotation sequence of the MacMini marker, where a poor registration when the camera was approaching parallel to the marker caused several erroneous registration calculations. OPIRA had a higher number of feature matches than the optical flow and standard implementations, often three or more times more matches were found using OPIRA. The OPIRA implementation had a higher number of successfully registered frames than both optical flow and standard implementations, in most cases increasing from below 50% for the standard implementation to above 65% for OPIRA.

Tables 7.16, 7.17 and 7.18 show the mean absolute error of the measured angle, average number of feature matches, and percent of successfully registered frames for the standard, optical flow and OPIRA implementations of the SURF registration algorithm. For most markers, the OPIRA implementation had a lower mean error than both the optical flow and standard implementations of the SURF algorithm. The less distinctive SURF registration algorithm had problems with the highly repetitive texture of the MacMini marker, which resulted in higher MAE using OPIRA, and the Grass marker,

	X			Y		
	S	OF	O	S	OF	O
MagicLand	19.87	15.88	13.08	7.77	18.13	3.40
Stop	48.52	44.88	30.05	33.68	26.02	20.79
Lucent	43.60	44.69	29.14	25.77	27.80	17.47
MacMini	6.60	15.51	16.65	6.53	27.15	4.72
Isetta	50.92	44.40	23.55	31.69	39.16	16.50
Philadelphia	30.84	36.28	9.21	25.14	28.92	19.25
Grass	36.21	21.85	13.64	21.86	25.63	11.91
Wall	28.11	45.67	26.30	13.42	26.95	4.93

Table 7.13: Mean absolute error for standard, optical flow and OPIRA implementations of the SIFT algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	29	29	103	37	35	105
Stop	16	18	35	18	19	38
Lucent	14	15	87	20	20	95
MacMini	24	21	215	36	30	234
Isetta	11	12	46	14	15	47
Philadelphia	27	25	141	31	31	135
Grass	6	7	46	7	8	49
Wall	23	21	121	33	31	135

Table 7.14: Average number of feature matches for standard, optical flow and OPIRA implementations of the SIFT algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	33%	39%	46%	52%	62%	83%
Stop	44%	61%	69%	45%	51%	66%
Lucent	38%	47%	73%	35%	37%	56%
MacMini	56%	58%	73%	54%	59%	72%
Isetta	33%	42%	74%	40%	41%	69%
Philadelphia	58%	51%	93%	61%	62%	77%
Grass	19%	34%	72%	33%	41%	63%
Wall	55%	54%	78%	62%	63%	87%

Table 7.15: Percentage of successfully registered frames for standard, optical flow and OPIRA implementations of the SIFT algorithm for the X and Y rotation sequences

which SURF failed completely to register. OPIRA had a higher number of feature matches than the optical flow and standard implementations for all markers but the low textured Stop marker, often three or more times more matches were found using OPIRA. The OPIRA implementation had a higher number of successfully registered frames than both optical flow and standard implementations for all markers but the low textured Stop marker, in most cases with an increase of approximately 25% successfully registered frames compared to the standard implementation.

Tables 7.19, 7.20 and 7.21 show the mean absolute error of the measured angle, average number of feature matches, and percent of successfully registered frames for the standard, optical flow and OPIRA implementations of the Ferns registration algorithm. The OPIRA implementation had a lower MAE for all markers compared to both the optical flow and standard implementations. OPIRA had at least twice as many feature matches compared to the optical flow and standard implementations for all markers. The OPIRA implementation had a higher number of successfully registered frames than both optical flow and standard implementations for all markers, in most cases with an increase of at least 20% successfully registered frames compared to the standard implementation.

7.2.3 Rotation Invariance

In Section 7.2.2, OPIRA was shown to add perspective invariance to the SIFT, SURF and Ferns algorithms when the camera was rotated about the X and Y axes. In this section, the improvements possible using OPIRA implementations of natural feature registration algorithms when the marker is rotated are empirically evaluated using the inertial orientation sensor ground truth image sequences captured in Section 7.1.5.

For clarity, the graphs in this section only show the difference between the rotation angle calculated using registration and measured using the ground truth. This difference is compared to the angle in rotation from perpendicular to the marker. As discussed in Section 7.1.5, the testing rig was rotated clockwise to 180° and then anti-clockwise to −180°, before returning to stationary.

	X			Y		
	S	OF	O	S	OF	O
MagicLand	8.10	10.93	37.85	5.22	18.74	11.57
Stop	36.21	29.29	50.07	34.70	23.14	30.47
Lucent	91.62	92.34	50.47	32.63	29.35	12.88
MacMini	9.85	22.68	27.01	5.74	23.85	13.63
Isetta	38.27	37.48	25.34	24.92	40.77	19.43
Philadelphia	23.09	37.78	9.56	9.55	32.68	14.45
Grass	∞	∞	∞	∞	∞	∞
Wall	17.58	39.02	10.43	7.34	17.25	4.81

Table 7.16: Mean absolute error for standard, optical flow and OPIRA implementations of the SURF algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	16	17	37	19	18	45
Stop	5	6	6	6	6	6
Lucent	5	5	12	6	7	32
MacMini	10	10	94	17	14	113
Isetta	10	9	28	10	10	27
Philadelphia	26	21	136	40	33	154
Grass	0	0	0	0	0	0
Wall	12	11	63	20	19	90

Table 7.17: Average number of feature matches for standard, optical flow and OPIRA implementations of the SURF algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	28%	35%	36%	36%	52%	60%
Stop	7%	16%	11%	12%	25%	18%
Lucent	1%	6%	33%	14%	25%	41%
MacMini	29%	35%	55%	45%	45%	62%
Isetta	25%	33%	61%	31%	36%	53%
Philadelphia	55%	57%	74%	54%	56%	82%
Grass	0%	0%	0%	0%	0%	0%
Wall	43%	39%	62%	57%	61%	69%

Table 7.18: Percentage of successfully registered frames for standard, optical flow and OPIRA implementations of the SURF algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	31.54	20.36	16.26	28.60	19.84	11.87
Stop	50.44	32.86	27.28	31.94	26.93	15.38
Lucent	44.40	37.00	25.79	30.85	25.54	22.47
MacMini	39.90	27.32	10.33	30.99	22.15	14.47
Isetta	41.10	29.27	20.54	26.88	26.14	10.46
Philadelphia	38.09	31.06	15.15	30.14	26.14	20.36
Grass	73.96	55.53	33.16	44.05	20.48	27.07
Wall	45.11	39.04	30.72	29.83	19.71	12.18

Table 7.19: Mean absolute error for standard, optical flow and OPIRA implementations of the Ferns algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	55	61	167	63	69	182
Stop	16	17	32	18	20	38
Lucent	32	35	125	41	44	143
MacMini	32	35	187	37	40	193
Isetta	26	27	72	31	33	75
Philadelphia	34	37	148	42	45	158
Grass	6	9	51	8	11	61
Wall	38	41	131	51	55	148

Table 7.20: Average number of feature matches for standard, optical flow and OPIRA implementations of the Ferns algorithm for the X and Y rotation sequences

	X			Y		
	S	OF	O	S	OF	O
MagicLand	34%	39%	46%	52%	66%	76%
Stop	39%	49%	64%	51%	53%	64%
Lucent	58%	61%	73%	42%	41%	52%
MacMini	58%	61%	84%	53%	59%	74%
Isetta	59%	61%	78%	53%	54%	76%
Philadelphia	59%	66%	82%	57%	64%	71%
Grass	18%	29%	63%	29%	45%	55%
Wall	57%	67%	77%	63%	71%	82%

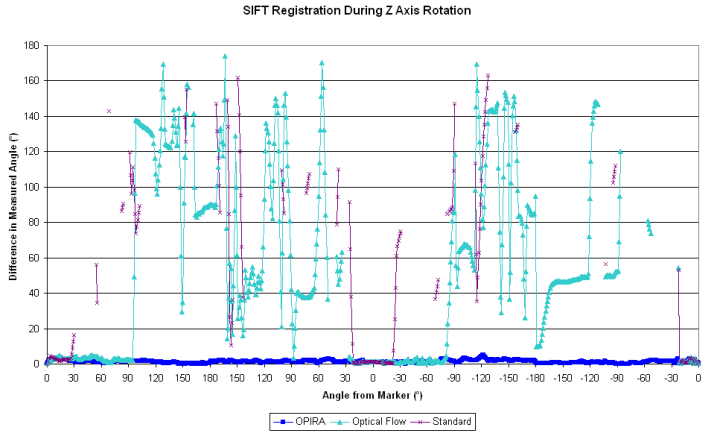
Table 7.21: Percentage of successfully registered frames for standard, optical flow and OPIRA implementations of the SURF algorithm for the X and Y rotation sequences

Figure 7.30 shows the differences between the rotation angle calculated using the three registration implementations and measured using the ground truth for SIFT, SURF and Ferns registration of the MagicLand marker. The standard registration implementations of the rotation dependent SIFT and SURF, shown in purple, fails to register after approximately 30° rotation in the clockwise and anti-clockwise direction. The optical flow implementations, shown in light blue, fail to register after approximately 90° rotation for SIFT, and 60° rotation for SURF. The OPIRA implementation, shown in dark blue, is rotation invariant and succeeds for all rotations. The Ferns algorithm is rotation independent by default, and all implementations succeed for all rotations.

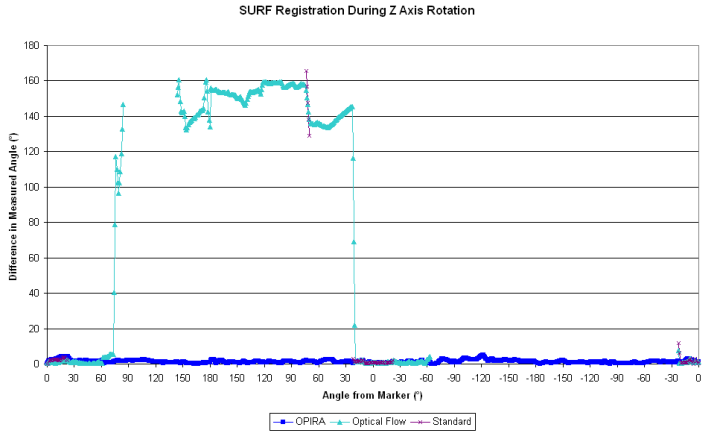
The published descriptions of the SIFT (Lowe 2004) and SURF (Bay et al. 2006) include rotation invariance by calculating an orientation when computing the descriptors for each feature. Figure 7.31 shows the difference between the measured angle and ground truth angle for SIFT (a) and SURF (b), with the standard and OPIRA implementations of the rotation invariant algorithms, and the OPIRA implementation of the rotation dependent algorithm. The overall accuracy of the algorithms is very good, with a maximum error of approximately 5° for SIFT and 6° for SURF.

Tables 7.22-7.27 compare the accuracy of the rotation dependent and rotation invariance SIFT and SURF algorithms using the standard, optical flow and OPIRA implementations. In this way, the rotation invariance provided by OPIRA can be compared to the rotation invariance implicit in the registration algorithms. As the Ferns algorithm is rotationally invariant by nature of its descriptor, it is exempt from this evaluation.

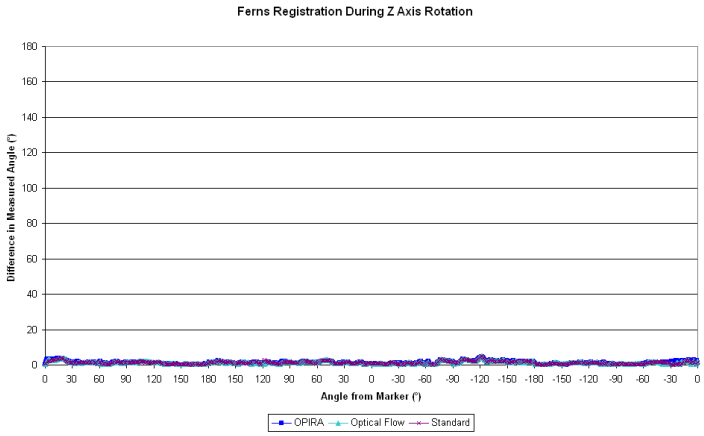
Tables 7.22, 7.23 and 7.24 show the mean absolute error of the measured angle, average number of feature matches, and percent of successfully registered frames for the standard, optical flow and OPIRA implementations of the rotation dependent and invariant SIFT registration algorithms. For rotation dependent SIFT, the OPIRA implementation had a far lower MAE than both the standard and optical flow implementations. For the rotation independent SIFT, there was little difference between the three implementations, except for the Grass marker, which the Rotation Invariant implementation failed to register due to high amounts of repetitive detail. OPIRA had a



(a) SIFT

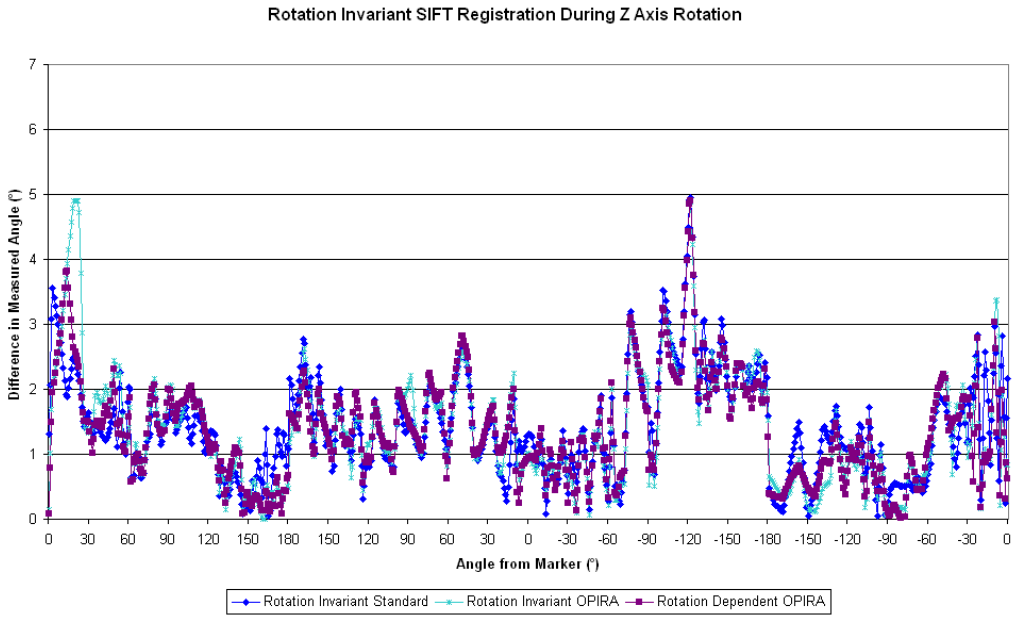


(b) SURF

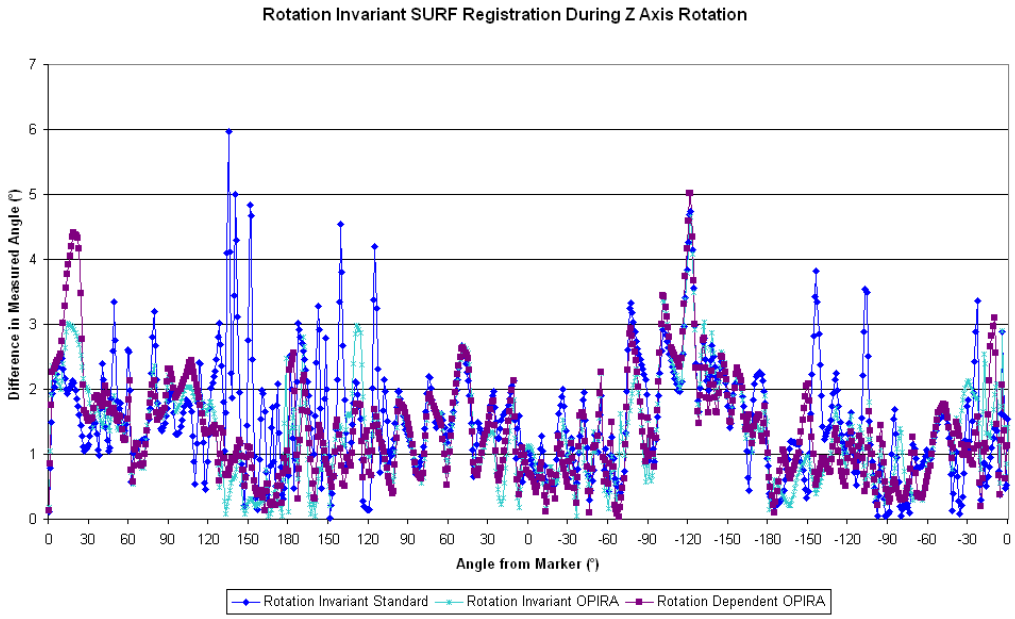


(c) Ferns

Figure 7.30: SIFT, SURF and Ferns registration results compared with the inertial orientation sensor ground truth during Z axis rotation for the MagicLand marker. Breaks in the line occur when there were not enough feature matches to complete registration



(a)



(b)

Figure 7.31: Rotation Invariant SIFT and SURF registration results compared with the inertial orientation sensor ground truth during Z axis rotation for the MagicLand marker

higher number of feature matches than the optical flow and standard implementations for both the rotation dependent and invariant SIFT algorithms. The OPIRA implementation had a far higher number of successfully registered frames than both optical flow and standard implementations for the rotation dependent SIFT, and was as good as or better than the implementations of rotation independent SIFT.

Tables 7.25, 7.26 and 7.27 show the mean absolute error of the measured angle, average number of feature matches, and percent of successfully registered frames for the standard, optical flow and OPIRA implementations of the rotation dependent and invariant SURF registration algorithms. For rotation dependent SURF, the OPIRA implementation had a far lower MAE than both the standard and optical flow implementations with the exception of the low textured Stop marker, and the Grass marker, where SURF failed. For the rotation independent SURF, the OPIRA algorithm outperformed on the repetitive textured Lucent and MacMini markers, and was approximately the same for the other markers, apart from the low textured Stop marker. OPIRA had a higher number of feature matches than the optical flow and standard implementations for both the rotation dependent and invariant SURF algorithms. The OPIRA implementation had a far higher number of successfully registered frames than both optical flow and standard implementations for the rotation dependent SURF, and was as good as or better than the implementations of rotation independent SURF.

These results show that the rotation invariance provided by OPIRA is, in almost all cases, as good as or better than the rotation invariant algorithms for MAE, number of point matches, and percentage of successfully registered frames.

7.2.4 Selection Process

As described in Chapter 5, OPIRA uses a best of three selection process to identify the source with the maximum number of feature matches; registration of the original image, optical flow tracking, or registration of the rectified image. As explained in Section 5.3.1, this approach is computationally inefficient, in most circumstances a single source will provide enough features

	Rot Dependent			Rot Invariant		
	S	OF	O	S	OF	O
MagicLand	49.72	46.54	1.10	1.10	1.22	1.17
Stop	94.50	92.94	2.06	2.04	2.10	2.06
Lucent	99.06	77.26	2.04	2.09	2.02	2.04
MacMini	58.37	84.99	2.21	2.17	2.19	2.22
Isetta	96.27	95.41	5.27	9.67	6.19	4.51
Philadelphia	91.24	66.50	2.75	2.71	1.94	1.93
Grass	51.11	47.18	2.07	62.02	21.58	6.25
Wall	94.86	69.48	1.85	1.86	1.87	1.85

Table 7.22: Mean absolute error for standard, optical flow and OPIRA implementations of the SIFT algorithm for the Z rotation sequence

	Rot Dependent			Rot Invariant		
	S	OF	O	S	OF	O
MagicLand	16	13	122	122	43	126
Stop	11	13	50	39	41	54
Lucent	11	11	137	27	31	124
MacMini	21	14	302	34	36	235
Isetta	8	10	67	17	20	56
Philadelphia	18	19	200	57	60	190
Grass	6	7	66	5	7	44
Wall	17	18	187	54	58	171

Table 7.23: Average number of feature matches for standard, optical flow and OPIRA implementations of the SIFT algorithm for the Z rotation sequence

	Rot Dependent			Rot Invariant		
	S	OF	O	S	OF	O
MagicLand	27%	42%	100%	100%	100%	100%
Stop	26%	31%	96%	96%	97%	97%
Lucent	27%	33%	98%	97%	98%	98%
MacMini	19%	38%	94%	96%	95%	95%
Isetta	27%	36%	97%	88%	94%	96%
Philadelphia	26%	36%	97%	98%	98%	97%
Grass	18%	22%	94%	36%	77%	91%
Wall	26%	38%	98%	97%	97%	98%

Table 7.24: Percentage of successfully registered frames for standard, optical flow and OPIRA implementations of the SIFT algorithm for the Z rotation sequence

	Rot Dependent			Rot Invariant		
	S	OF	O	S	OF	O
MagicLand	10.04	62.16	1.12	1.12	1.24	1.11
Stop	60.51	104.77	96.81	63.41	96.56	130.79
Lucent	87.41	96.74	4.32	99.40	77.30	3.76
MacMini	25.21	94.51	2.22	28.73	4.29	2.24
Isetta	91.53	111.51	5.24	6.49	9.40	4.46
Philadelphia	86.06	71.31	2.74	1.92	1.93	2.74
Grass	∞	∞	∞	∞	∞	∞
Wall	56.58	73.73	1.84	2.27	1.86	1.84

Table 7.25: Mean absolute error for standard, optical flow and OPIRA implementations of the SURF algorithm for the Z rotation sequence

	Rot Dependent			Rot Invariant		
	S	OF	O	S	OF	O
MagicLand	13	9	51	51	11	42
Stop	5	5	6	4	4	5
Lucent	5	5	43	4	5	30
MacMini	13	8	170	10	12	107
Isetta	7	8	43	13	14	38
Philadelphia	17	19	224	40	43	191
Grass	0	0	0	0	0	0
Wall	17	11	116	17	19	92

Table 7.26: Average number of feature matches for standard, optical flow and OPIRA implementations of the SURF algorithm for the Z rotation sequence

	Rot Dependent			Rot Invariant		
	S	OF	O	S	OF	O
MagicLand	25%	36%	100%	100%	99%	100%
Stop	19%	23%	32%	1%	0%	3%
Lucent	5%	7%	94%	5%	5%	94%
MacMini	19%	35%	94%	72%	91%	94%
Isetta	26%	31%	96%	94%	93%	97%
Philadelphia	28%	37%	97%	97%	97%	97%
Grass	0%	0%	0%	0%	0%	0%
Wall	27%	41%	98%	98%	99%	98%

Table 7.27: Percentage of successfully registered frames for standard, optical flow and OPIRA implementations of the SURF algorithm for the Z rotation sequence

for an accurate registration.

To evaluate the efficiency of the selection process, a video sequence was captured of rotation around the X axis of the marker, followed by rotation around the Y axis of the marker, as shown in Figure 7.32. The number of feature matches found by the three sources of the OPIRA implementation of SURF was measured.

Figure 7.33 shows the number of successfully registered features from each source at each frame. After initial registration, registration of the original image finds the lowest number of features matches. As a result, the OPIRA selection process will never select this method, and it is unnecessary unless reinitialisation is required. The number of feature matches found by optical flow and OPIRA are very similar.

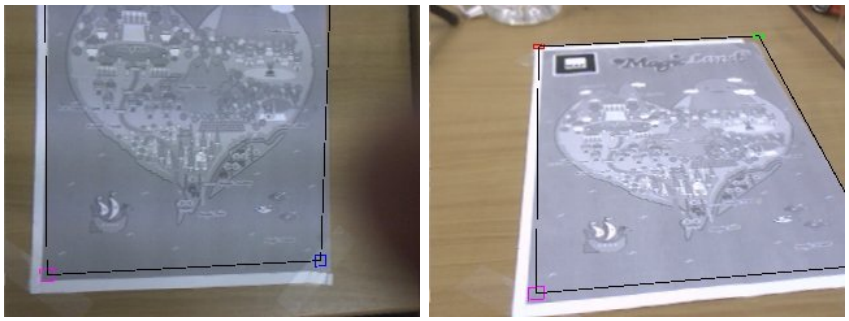
As the camera approaches parallel to the marker in frames 65 to 105, optical flow is primarily used. As perspective distortion of the marker increases, the number of feature matches tracked by optical flow reduces at a slower rate than those found using registration of the rectified image. As the camera returns to a position perpendicular to the marker in frames 105 to 121, perspective distortion decreases. This increases the detail available for registration, increasing the feature matches found in the registration of the rectified image.

As shown in Figure 7.32, there is tracking failure from frames 166 to 180. At frame 181, registration of the rectified image finds enough reliable features to correct this error in registration without need for re-initialisation.

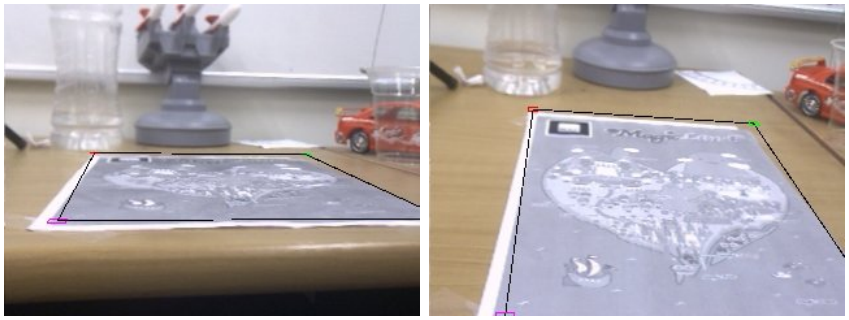
Figure 7.34 shows the source selected using the best of three selection process during OPIRA registration of the video sequence. Registration of the original image is only used for initialisation, afterwards the source varies between registration of the rectified image and optical flow.

7.3 Blur Invariance

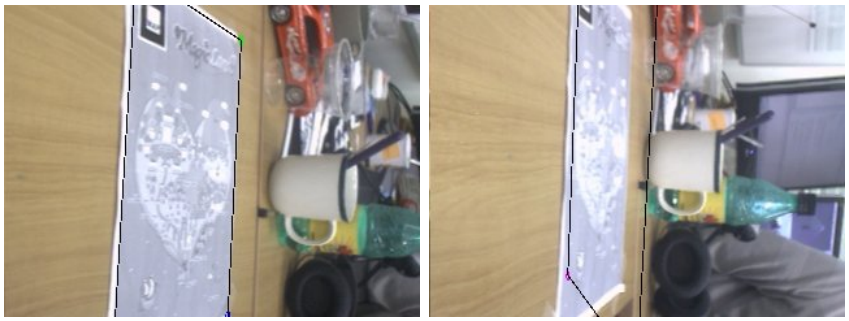
The accuracy of registration is dependent on the quality of the image. With a properly calibrated camera, the most common source of image degradation is noise. In this research noise is divided into two categories, local noise which corrupts parts of the image (Section 6.1.1) and global noise, which affects the



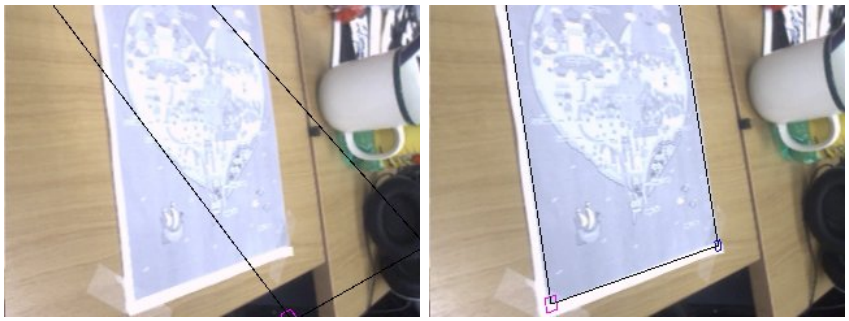
(a) Frame 000 (b) Frame 075



(c) Frame 105 (d) Frame 112



(e) Frame 154 (f) Frame 166



(g) Frame 180 (h) Frame 181

Figure 7.32: A frame sequence involving rotating a camera around a stationary marker in the X and Y axes. The black bounding box is drawn according to the calculated extrinsic parameters. Registration failure can be seen in frame 166 and 180 (Clark et al. 2008)

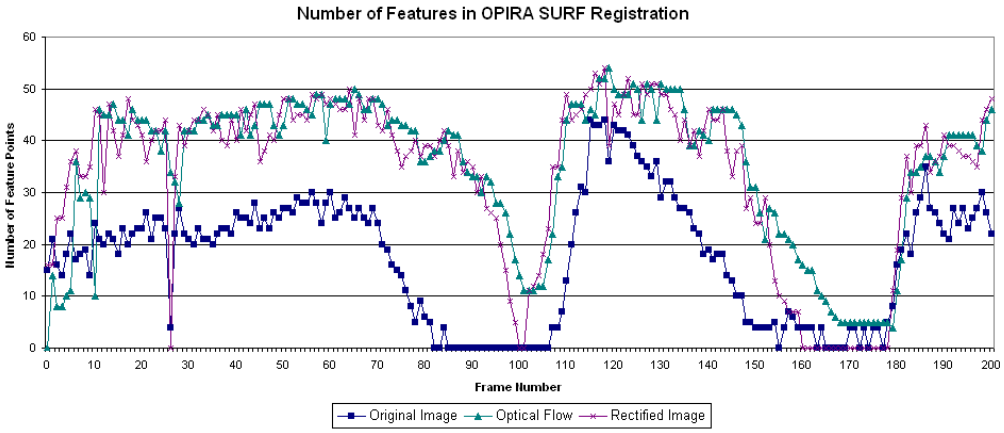


Figure 7.33: The number of feature matches detected using registration of the original image, optical flow, and registration of the rectified image using the OPIRA implementation of SURF (Clark et al. 2008)

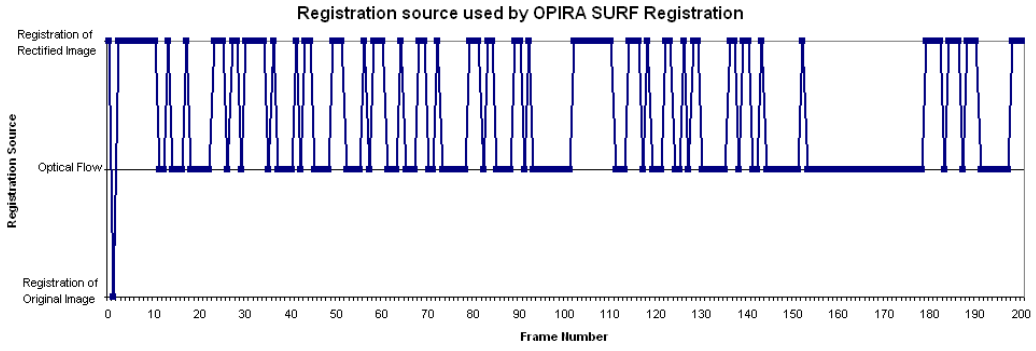


Figure 7.34: The registration source chosen due to the highest number of feature matches (Clark et al. 2008)

image as a whole (Sections 6.1.2).

Local noise is pervasive in computer vision, and all natural feature registration algorithms provide a degree of invariance to this problem. Any features corrupted due to the presence of local noise are usually removed during the feature matching or RANSAC homography estimation. Conversely, natural feature registration algorithms are susceptible to the problem of global noise such as blur, and registration accuracy can be improved with filters to remove these weaknesses.

In Section 6.1.2, blur is subdivided into two categories, out-of-focus blur and motion blur. In the context of blur removal, the only difference between these two categories is the point spread function (PSF) used in the deconvolution process. As such, the impact of blur removal on registration is the same regardless of the source of blur.

(a) (b) (c)

Figure 7.35: Image convolution, (a) The source image, (b) The PSF of the defocused camera, (c) The resulting blurred image

As neither out-of-focus nor motion blur was present in the rotation sequence videos, a convolution function was used to introduce the problem. The effect of out-of-focus blur can be more accurately emulated than the effect of motion blur as the PSF for out-of-focus blur is static. Out-of-focus blur has been shown to reduce registration for planar fiducial markers by Zhang, Fronz and Navab (2002).

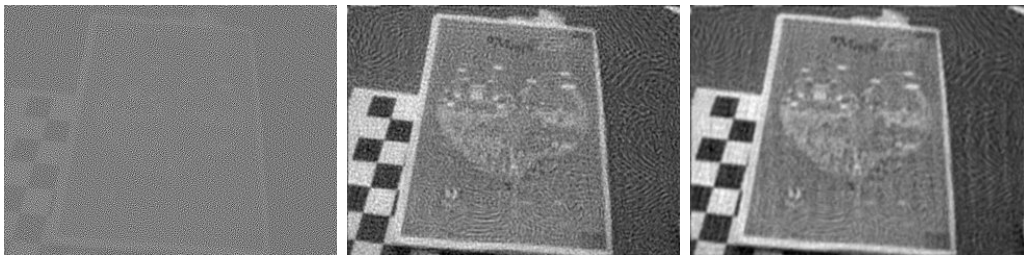
The PSF for out-of-focus blur was obtained by placing the camera in a sealed box with a pinhole aperture. Light was shone through the pinhole into the camera lens, and the camera was defocused until the point of light

became a disk. Figure 7.35 shows the disk of light equivalent to the point spread function of the blur (b), and an example image before (a) and after (c) convolution with the point spread function.

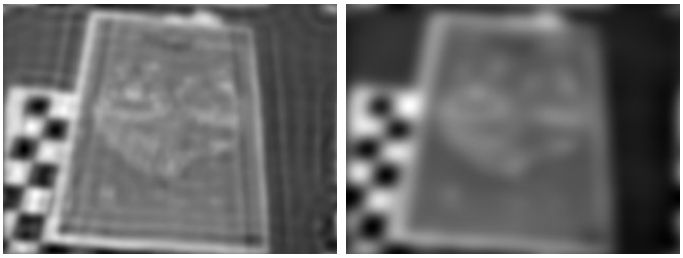
The deconvolution algorithm chosen for evaluation was the Wiener filter, shown in Algorithm 6.1.1. The Wiener filter has three parameters; Gamma, SNR and threshold, which are calibrated in the following section.

7.3.1 Parameter Calibration

The first Wiener filter parameter calibrated was the threshold.



(a) t=0.00000001 (b) t=0.00008192 (c) t=0.00032768



(d) t=0.02097152 (e) t=0.67108864

Figure 7.36: The results of Wiener deconvolution across a range of threshold values.

Threshold

The threshold represents the lowest frequency that the filter works at. If the threshold is set too low, the resulting image will be noisy and if set too high blur will still be present in the image. The SNR and Gamma parameters were set to zero, and results were recorded for threshold values of 0, and

incrementally doubled values between 1.0×10^{-8} and 1. Figure 7.36 shows the results of the deconvolution at 1.00×10^{-8}, 8.19×10^{-5}, 3.27×10^{-4}, 2.09×10^{-2} and 6.71×10^{-1}.

To quantify the effect of the Wiener Filter threshold, the number of feature matches found in the MagicLand marker using the OPIRA implementation of SIFT was measured for each filtered frame, as shown in Figure 7.37. As a baseline, the number of features found in the marker before blur and filtering was 19, and no features were found in the blurred marker with no filtering.

When the threshold was below 4.10×10^{-5} there was too much noise present for accurate registration. The maximum number of feature matches found was 19, the same number found in the image before blurring, at a threshold value of approximately 6.55×10^{-4}. After this peak, the number of feature matches decreased until a threshold value of 1.05×10^{-2} where the marker was too blurred for accurate registration.

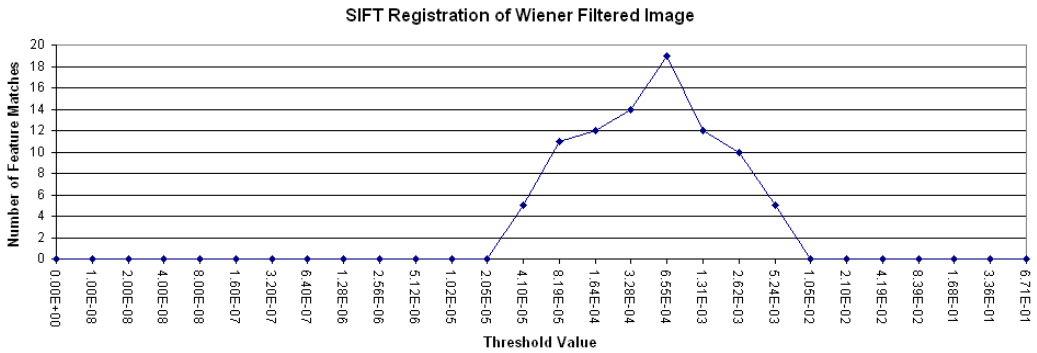


Figure 7.37: The number of feature matches found in the MagicLand marker compared to the threshold value. Registration of the original marker yielded 19 feature matches

SNR and Gamma

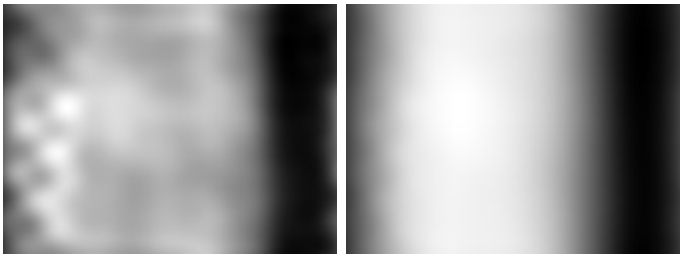
As shown in Algorithm 6.1.1, the SNR and Gamma parameters are interdependent due to their use in the equation:

$$Sm = Sf/(Sh \times Sf + Gamma \times SNR) \tag{7.4}$$

Due to this interdependence, it is unnecessary to test both variables independently. Instead, the effect that the product of the two variables has on the deconvolution can be evaluated.

The threshold parameter was set to 6.55×10^{-4}, the optimal value found in the previous experiment, and the results were recorded for the $SNR \times Gamma$ values of 0, and incrementally doubled values between 1×10^{-12} and 1 on the MagicLand marker. The results of deconvolution are shown in figure 7.38. Lower values of $SNR \times Gamma$ produce a better image, while higher values result in a decrease in the quality of the blur removal, ultimately leading to further degradation of the marker.

(a) s=0.000000000001 (b) s=0.000000001024 (c) s=0.000000524288



(d) s=0.000536870912 (e) s=0.549755813888

Figure 7.38: The results of Wiener deconvolution of the MagicLand marker across a range of $SNR \times Gamma$ values.

Figure 7.39 shows the number of feature matches found in the MagicLand marker compared to the $SNR \times Gamma$ value. The maximum number of feature matches found was 19 at a $SNR \times Gamma$ value of 0. After this peak, the number of feature matches decreases until a $SNR \times Gamma$ of 1.02×10^{-9} when the marker is too blurred for accurate registration.

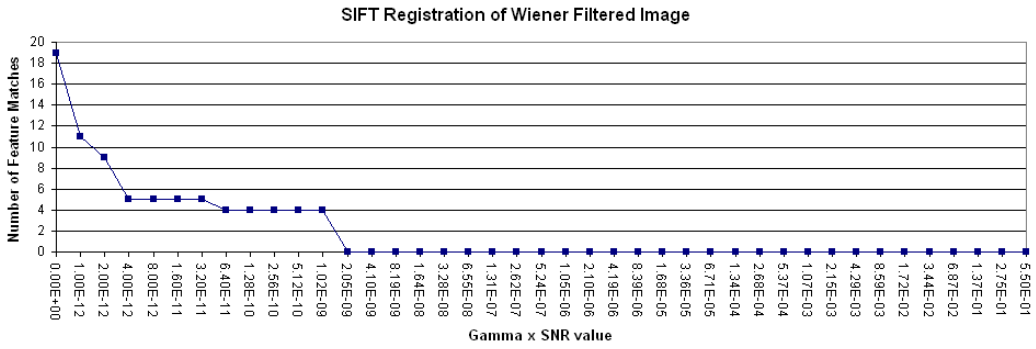


Figure 7.39: The number of feature matches found in the MagicLand marker compared to the $SNR \times Gamma$ value. Registration of the original marker yielded 19 feature matches

In a natural feature registration application, this parameter estimation process can be performed during the camera calibration stage. With the optimal values for the Gamma, SNR and threshold parameters, the effectiveness of blur removal for natural feature registration can be evaluated.

7.3.2 Evaluation

With the optimal values for the parameters of the Wiener filter known, the efficacy of deconvolution for blur removal to improve registration can be examined by varying the level of blur on each marker's video sequences, and comparing the registration accuracy before and after the filter has removed the blur.

As the point spread function is a measure of light dispersion related to focal distance of the lens, if the distortion effects of the lens are not altered when the focal distance is changed, resizing the point spread function is equivalent to changing the focal distance of the lens. The original point

spread function obtained in Figure 7.35(b) was digitally resized to generate five levels of blur, as shown in Figure 7.40. The functions range from a circle of distortion diameter of approximately 2 pixels (a), to approximately 38 pixels (e).

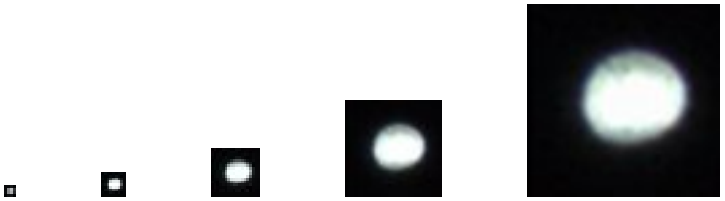


Figure 7.40: The point spread functions used for the evaluation

Figure 7.41 shows the result of the different levels of blur on the MagicLand marker and the results after the Wiener filter was applied. The first three filtered markers have a high level of detail visible, although the final two images were too corrupted for complete restoration using the Wiener filter.

The performance of the standard, optical flow and OPIRA implementations of SIFT, SURF and Ferns were evaluated for the MagicLand marker video sequences, and the results are shown in Figure 7.42. The three implementations have a strong correlation with the decrease in the percentage of successfully registered frames as blur increases. As OPIRA had the best performance in this evaluation and was shown to provide the lowest error and highest percentage of successfully registered frames for all markers and video sequences in Section 7.2, the OPIRA implementations of the SIFT, SURF and Ferns registration algorithms were used for the evaluation of the Wiener filter across the other markers.

Each markers X, Y and Z rotation sequence was artifically blurred using the PSFs shown in Figure 7.40, and registration was performed as a baseline. The blurred video sequences were filtered using the Wiener filter, and registration was performed again to evaluate the improvements.

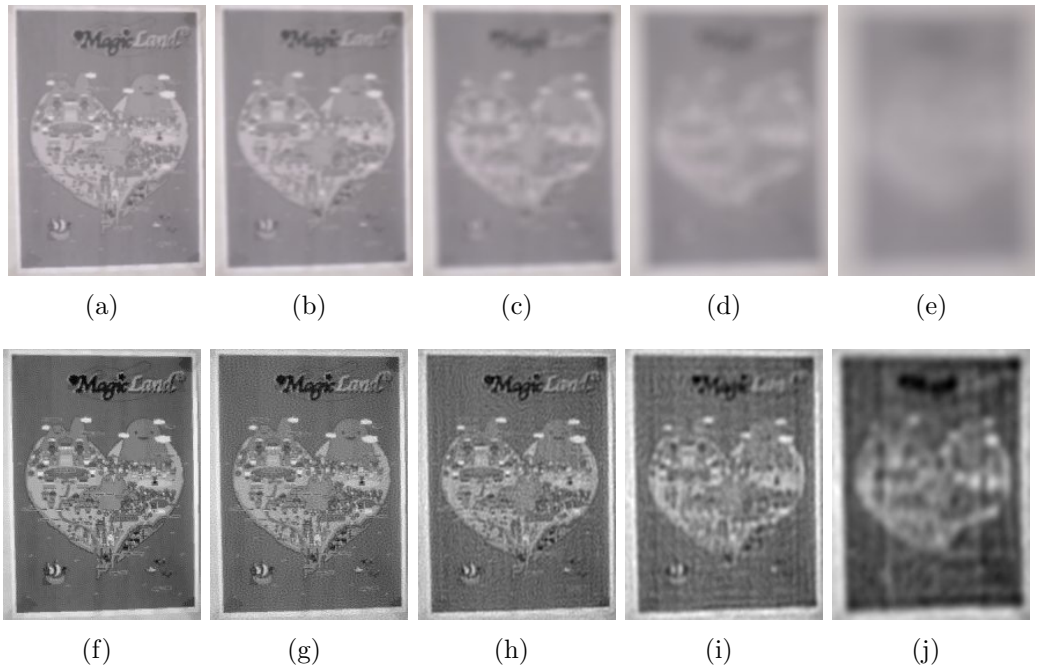
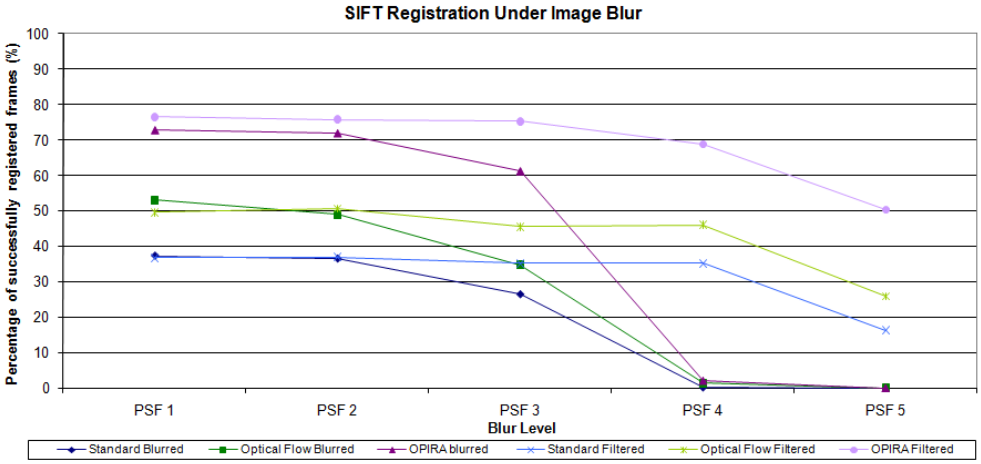


Figure 7.41: The MagicLand marker convolved with the different point spread functions (top), and the Wiener deconvolved images (bottom)

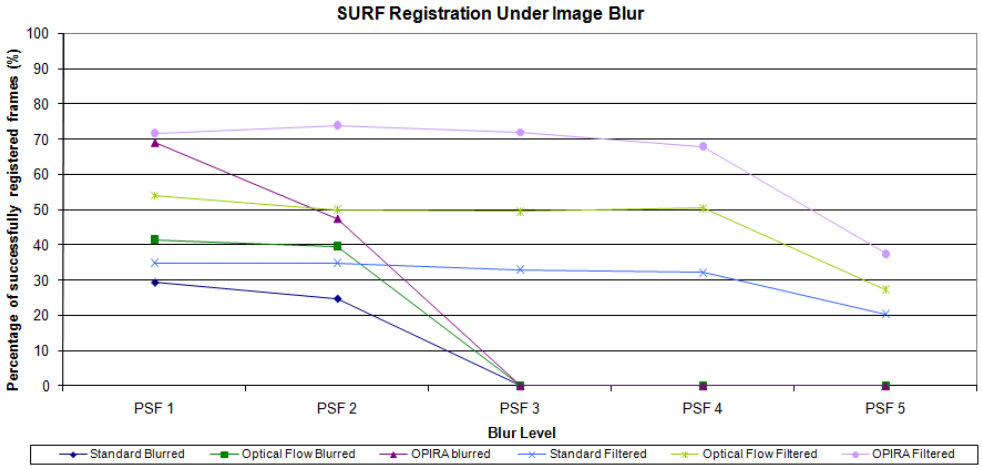
SIFT Table 7.28 shows an increase in the mean absolute error with an increase with the scale of blur for both the blurred and filtered image sequences. The error increases at a slower rate in the filtered images than the blurred ones, and most filtered sequences have a lower error than the corresponding blurred sequence. At the highest level of blur, the error is higher for repetitive or highly textured markers such as the MacMini and Grass markers.

These results correlate to the number of feature matches as shown in Table 7.29. An increase in the scale of blur causes a decrease in the number of feature matches. However, after using the Wiener filter, this decrease occurs at a slower rate, and only after the fourth scale do the number of features drop below 20 for most markers.

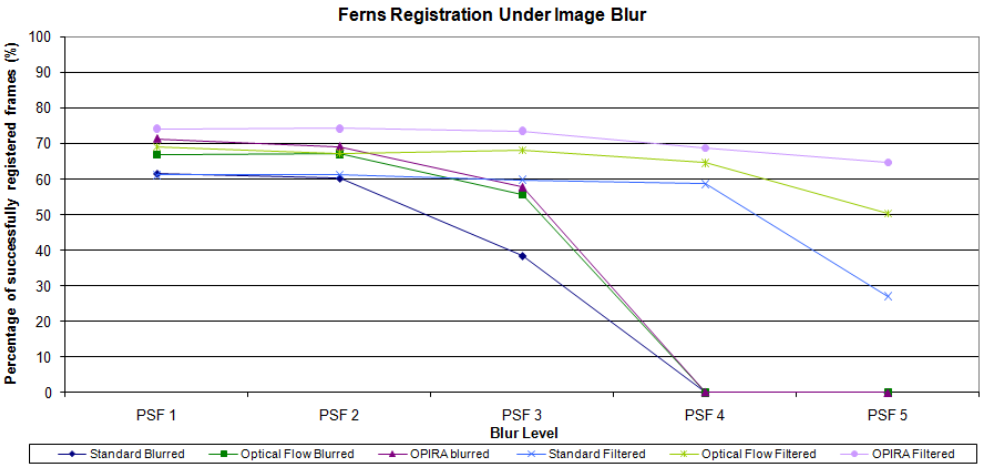
The percentage of successfully registered frames also decreases as the scale of blur increases, as shown in Table 7.30. Up to the fourth scale of blur, all marker sequences apart from Grass have over 60% registration success after



(a) SIFT



(b) SURF



(c) Ferns

Figure 7.42: The percentage of successful registrations for the SIFT, SURF and Ferns algorithms

filtering, while all the blurred sequences drop below 50% registration success after the third scale of blur.

SURF Table 7.31 shows an increase in the mean absolute error with an increase with the scale of blur for both the blurred and filtered image sequences. All marker sequences fail at blur scales of 4 and above, with the MagicLand and MacMini failing at scale 3. As in previous experiments, SURF was unable to register the grass sequence at all. After filtering, SURF is able to register to a blur scale of 5, with the error decreasing at a significantly slower rate.

These results correlate to the number of feature matches as shown in Table 7.32. An increase in the scale of blur causes a decrease in the number of feature matches. However, after using the Wiener filter, this decrease occurs at a much slower rate.

The percentage of successfully registered frames also decreases as the scale of blur increases, as shown in Table 7.33. Up to the fourth scale of blur, almost all marker sequences have over 50% registration success after filtering, while all the blurred sequences drop below 50% registration success after the second scale of blur.

Ferns Table 7.34 shows an increase in the mean absolute error with an increase with the scale of blur for both the blurred and filtered image sequences. The error increases at a slower rate in the filtered images than the blurred ones, and all filtered sequences have a lower error than the corresponding blurred sequence, with the exception of the MacMini marker sequence at the first scale of blur. The error at the highest level of blur is below 45 for all sequences but the MacMini, Grass and Wall, which have high and repetitive textures.

These results correlate to the number of feature matches as shown in Table 7.35. An increase in the scale of blur causes a decrease in the number of feature matches. However, after using the Wiener filter, this decrease occurs at a slower rate, and only after the fourth scale do the number of features drop below 20 for all but the Grass marker.

The percentage of successfully registered frames also decreases as the

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	9.01	7.88	12.1	85.36	∞	6.38	5.89	6.48	16.97	29.26
S	20.96	24.36	28.75	84.74	87.19	21.40	21.63	23.54	26.12	46.41
L	20.07	27.10	35.74	66.42	36.16	15.57	19.53	20.27	21.50	59.77
MM	15.82	20.27	18.33	66.15	76.20	8.06	12.50	8.12	22.74	78.56
I	12.91	18.45	33.06	81.40	84.76	16.90	16.19	18.06	22.80	37.16
P	21.41	20.15	33.14	80.33	89.30	12.55	10.59	14.37	19.64	39.87
G	14.72	16.23	68.48	85.87	82.49	11.21	11.26	8.32	70.53	80.35
W	9.23	21.05	17.77	61.74	69.52	13.53	9.03	14.91	16.56	30.30

Table 7.28: Mean absolute error of the SIFT algorithm for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	103	77	25	2	0	108	99	90	78	20
S	38	29	12	5	5	38	34	30	25	11
L	94	52	14	5	4	102	92	76	49	8
MM	227	118	15	4	4	228	217	208	96	5
I	49	34	14	5	4	50	45	45	36	13
P	125	67	15	5	4	150	148	137	82	13
G	55	30	5	4	4	51	37	30	7	5
W	117	64	22	4	4	140	139	123	70	13

Table 7.29: Average number of feature matches found by the SIFT algorithm for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	73%	72%	61%	35%	16%	77%	76%	75%	69%	50%
S	72%	68%	52%	5%	1%	73%	73%	71%	60%	30%
L	68%	67%	53%	12%	2%	70%	72%	67%	61%	17%
MM	77%	75%	61%	3%	0%	82%	79%	76%	72%	12%
I	75%	71%	53%	3%	2%	80%	77%	71%	67%	43%
P	79%	75%	56%	4%	0%	84%	89%	83%	79%	42%
G	70%	65%	6%	0%	0%	70%	67%	60%	18%	4%
W	87%	80%	69%	3%	0%	84%	84%	83%	79%	52%

Table 7.30: Percentage of successfully registered frames by the SIFT algorithm for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	6.9	20.43	∞	∞	∞	8.28	5.02	7.77	11.44	34.58
S	53.82	50.13	68.76	∞	∞	21.89	28.73	18.66	33.25	51.03
L	31.28	49.82	108.64	∞	∞	23.97	23.18	17.72	21.81	72.54
MM	15.89	20.14	∞	∞	∞	16.90	13.52	12.37	10.62	62.50
I	28.63	26.06	54.22	168.52	∞	15.23	16.08	19.33	23.94	52.63
P	5.89	14.15	37.31	∞	∞	12.28	9.18	8.35	20.53	30.12
G	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
W	8.22	16.95	279.47	∞	∞	7.38	7.85	6.76	12.27	26.49

Table 7.31: Mean absolute error of the SURF algorithm for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	30	10	0	0	0	85	77	71	54	9
S	5	5	4	0	0	11	8	9	8	5
L	19	6	4	0	0	37	32	31	19	5
MM	74	15	0	0	0	152	129	115	54	4
I	24	12	5	4	0	38	35	36	28	10
P	124	52	10	0	0	176	166	146	91	17
G	0	0	0	0	0	0	0	0	0	0
W	55	14	4	0	0	118	107	98	56	8

Table 7.32: Average number of feature matches found by the SURF algorithm for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	69%	47%	0%	0%	0%	72%	74%	72%	68%	37%
S	20%	8%	2%	0%	0%	53%	46%	49%	47%	16%
L	42%	18%	0%	0%	0%	53%	50%	52%	56%	5%
MM	60%	49%	0%	0%	0%	66%	66%	65%	64%	1%
I	66%	51%	15%	0%	0%	70%	71%	69%	66%	34%
P	79%	69%	34%	0%	0%	84%	85%	86%	74%	44%
G	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
W	66%	45%	0%	0%	0%	78%	78%	76%	75%	34%

Table 7.33: Percentage of successfully registered frames by the SURF algorithm for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	10.72	17.31	19.92	∞	∞	6.99	7.95	8.69	15.53	21.59
S	21.33	25.68	34.37	81.91	73.70	20.62	16.26	23.77	24.06	33.92
L	23.88	25.05	33.35	78.71	76.98	18.68	16.51	23.37	26.71	37.15
MM	12.00	20.96	36.81	91.30	88.04	16.34	14.86	19.39	24.85	70.68
I	17.58	25.16	32.67	80.55	91.68	13.67	18.30	16.62	22.08	33.80
P	19.03	19.07	41.43	84.18	82.59	13.21	16.63	15.19	17.65	42.04
G	30.33	60.74	99.21	94.81	96.71	25.19	25.17	26.40	84.03	80.82
W	16.29	23.75	40.06	90.12	70.98	14.05	18.72	18.12	23.81	52.22

Table 7.34: Mean absolute error of the Ferns classifier for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	148	73	15	1	0	225	219	194	119	27
S	31	18	8	5	4	47	45	42	29	12
L	117	42	8	5	4	163	153	128	55	10
MM	183	80	8	4	4	219	207	174	74	6
I	60	30	11	5	4	92	88	81	47	16
P	136	62	12	5	4	178	172	149	76	14
G	29	6	5	5	4	83	72	50	5	5
W	127	66	11	5	4	168	162	146	82	9

Table 7.35: Average number of feature matches found by the Ferns classifier for each marker after blurring and filtering at five point spread function scales

	Blurred (PSF #)					Filtered (PSF #)				
	1	2	3	4	5	1	2	3	4	5
ML	71%	69%	58%	0%	0%	74%	74%	74%	69%	65%
S	68%	62%	39%	3%	7%	73%	75%	72%	65%	43%
L	67%	66%	41%	3%	6%	74%	72%	68%	63%	39%
MM	77%	73%	40%	5%	5%	79%	82%	77%	71%	13%
I	76%	68%	53%	7%	2%	83%	77%	78%	71%	51%
P	77%	75%	47%	6%	4%	85%	83%	81%	76%	40%
G	59%	17%	4%	4%	3%	69%	68%	62%	5%	3%
W	80%	76%	45%	2%	6%	86%	82%	82%	75%	32%

Table 7.36: Percentage of successfully registered frames by the Ferns classifier for each marker after blurring and filtering at five point spread function scales

scale of blur increases, as shown in Table 7.36. Up to the fourth scale of blur, all marker sequences except Grass have over 60% registration success after filtering, while all the blurred sequences drop below 50% registration success after the second scale of blur.

Overall results The rate of decrease of the percentage of successfully registered frames was reduced by using the Wiener filter for all registration algorithms. The SIFT registration algorithm, which was the most robust to blur, increased the average percentage of successfully registered frames from 50% at the third scale of blur to 60% at the fourth scale after the Wiener filter was applied. The SURF algorithm, which was least robust to blur, increased the average percentage of successfully registered frames from 50% at the second scale of blur to 60% at the fourth scale after the Wiener filter was applied. The Ferns classifier increased the average percentage of successfully registered frames from 45% at the third scale of blur to 65% at the fourth scale after the Wiener filter was applied.

By using the Wiener filter, the robustness of all the registration algorithms to blur was improved. However, the results still show a decline in the error, number of feature matches, and percentage of successfully registered frames as the blur level increases. At extremely high levels of blur, the image will be too corrupt for the Wiener filter to recover accurate information from for registration.

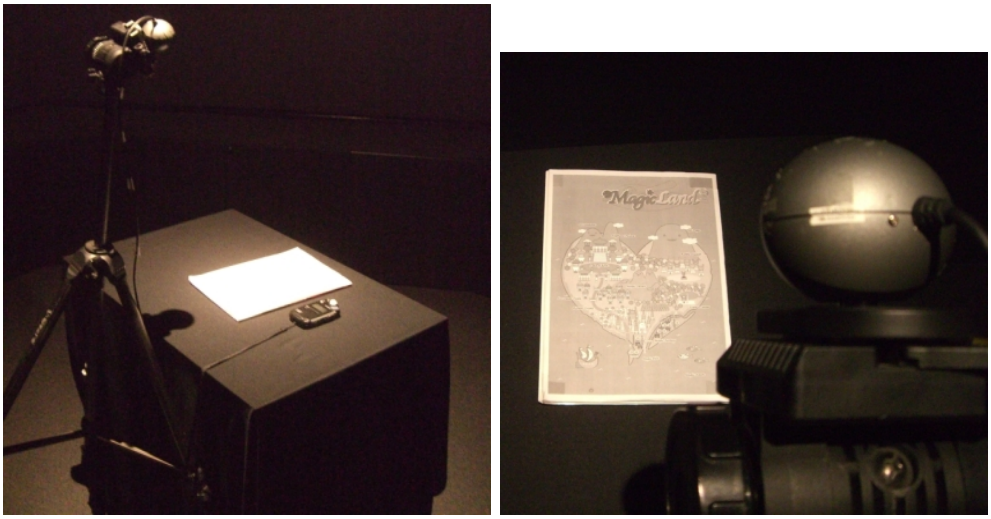
7.4 Illumination Invariance

Many natural feature registration algorithms use intensity based algorithms for calculation of the feature descriptors. A decrease in the ambient illumination results in a decrease in the variance of intensity within an environment. A decrease in the variance of intensity decreases the uniqueness of feature descriptors, reducing the accuracy of registration.

Histogram equalisation is a computationally efficient and invertible operation to increase the variance in intensity, as described in Section 6.2. The intensity values for each pixel in the image are used to create a histogram, which is stretched so the distribution covers the entire spectrum of intensi-

ties, as shown in Figure 6.6. This improves the visibility of objects in poorly lit images, while not affecting well lit images.

To evaluate the efficiency of histogram equalisation as a method of improving the accuracy of natural feature registration, a test environment was set up where the ambient light levels could be precisely controlled. The only light source was an adjustable incandescent illuminant located directly above the marker, and all surfaces were covered with black felt to minimise the amount of reflected light. The light level was measured with a Sekonic Flashmate L-308S light meter[7]. The test environment is shown in Figure 7.43.



(a) (b)

Figure 7.43: The experimental set-up for the histogram equalisation evaluation

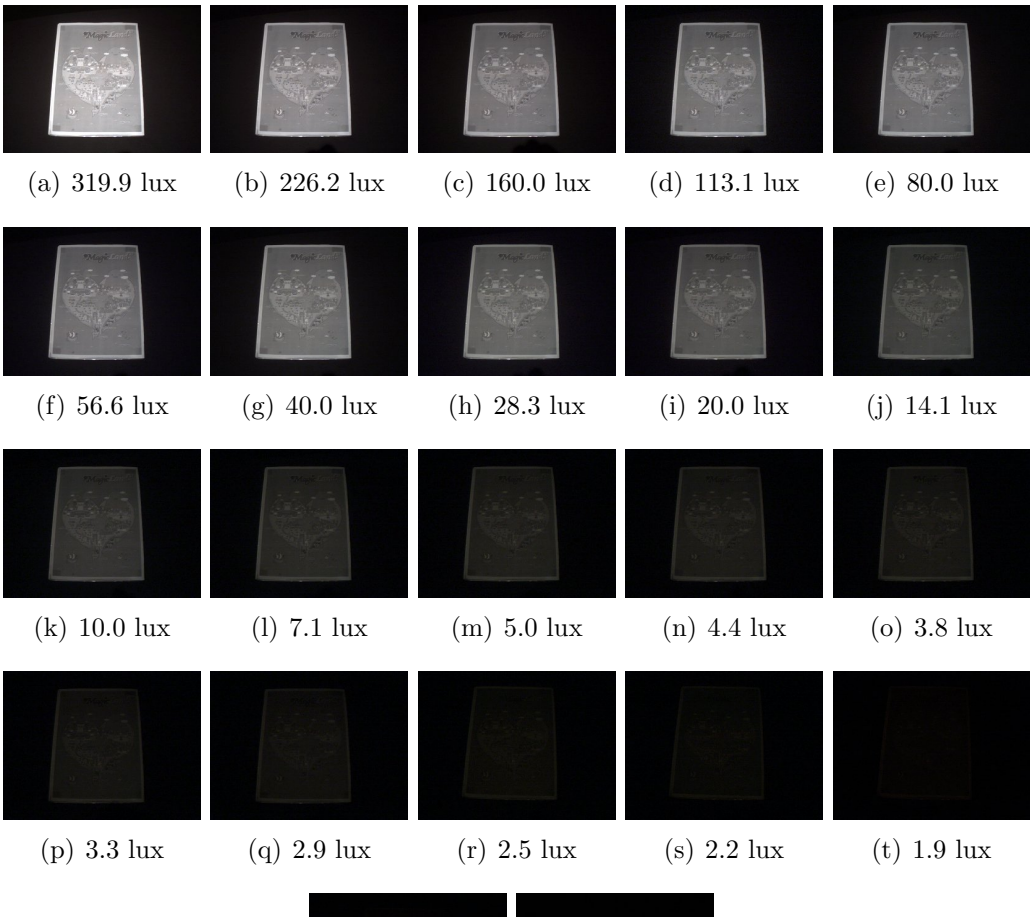
The Sekonic Flashmate measures light intensity in EV, a linear scale of light intensity, which is converted to lux using the formula:

$$Lux = 2.5e^{(0.6931 \times EV)} \tag{7.5}$$

Images were captured at discrete intervals for testing, starting at the typ-

[7] http://www.sekonic.com/products/Sekonic%20L-308S%20FLASHMATE.asp

ical office lighting level of 320 lux[8]. The light was decreased at steps of 5EV down to 5 lux. The increased percentage of difference between illumination levels at low light was captured by reducing the illumination by 0.2EV after 5 lux. The images captured for the MagicLand marker are shown in Figure 7.44.



(a) 319.9 lux (b) 226.2 lux (c) 160.0 lux (d) 113.1 lux (e) 80.0 lux

(f) 56.6 lux (g) 40.0 lux (h) 28.3 lux (i) 20.0 lux (j) 14.1 lux

(k) 10.0 lux (l) 7.1 lux (m) 5.0 lux (n) 4.4 lux (o) 3.8 lux

(p) 3.3 lux (q) 2.9 lux (r) 2.5 lux (s) 2.2 lux (t) 1.9 lux

(u) 1.6 lux (v) 0 lux

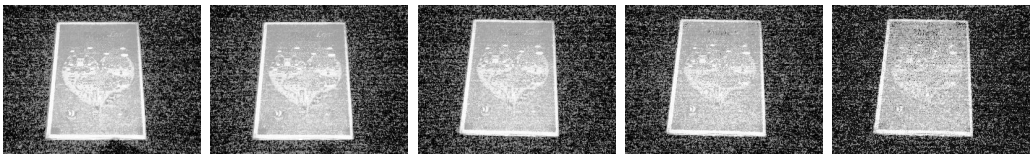
Figure 7.44: The captured images for the MagicLand marker before histogram equalisation

[8] Australian Standard Lux Level AS1680.2.4:1997 Table E1

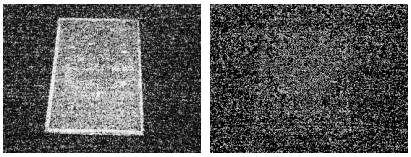
(a) 319.9 lux (b) 226.2 lux (c) 160.0 lux (d) 113.1 lux (e) 80.0 lux

(f) 56.6 lux (g) 40.0 lux (h) 28.3 lux (i) 20.0 lux (j) 14.1 lux

(k) 10.0 lux (l) 7.1 lux (m) 5.0 lux (n) 4.4 lux (o) 3.8 lux



(p) 3.3 lux (q) 2.9 lux (r) 2.5 lux (s) 2.2 lux (t) 1.9 lux



(u) 1.6 lux (v) 0 lux

Figure 7.45: The captured images for the MagicLand marker after histogram equalisation

Each marker was equalised to increase the variance of illumination. The MagicLand marker image sequence after equalisation is shown in Figure 7.45. Fine detail in this marker's image sequence becomes difficult to see below 14 lux in the unequalised images, with only faint detail visible below 4 lux. After histogram equalisation, fine detail is visible down to 3.8 lux, with limited visibility at 1.6 lux.

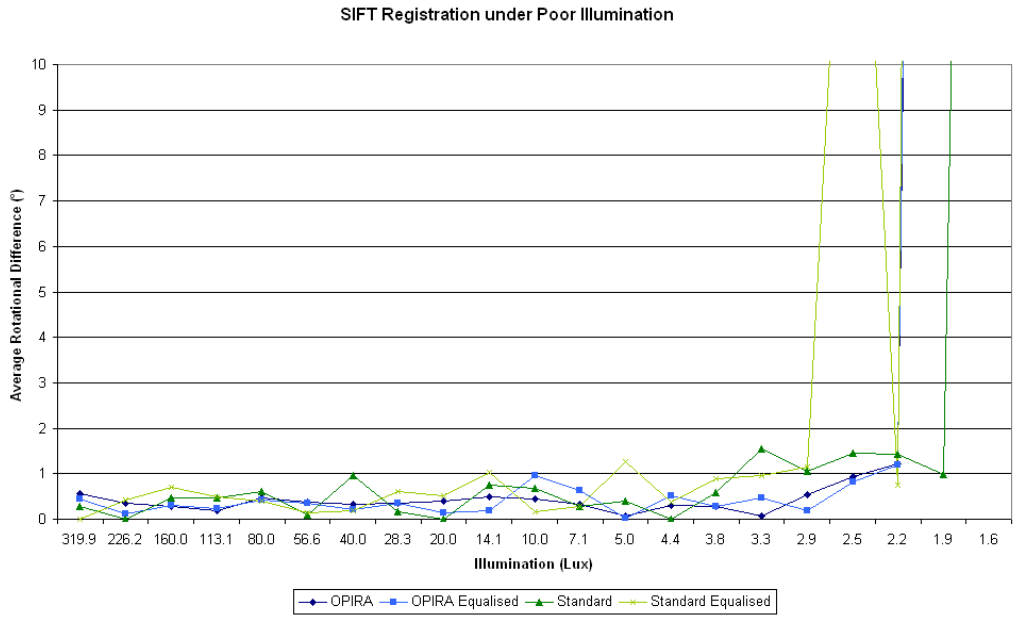
As the images captured were from a fixed camera, the inertial ground truth was of no use. Instead, a ground truth was calculated from the average of several SIFT registration matrices calculated for the marker at 320 lux.

As there was no motion in the camera, the optical flow implementation would not provide any additional benefit over the standard implementation. The performance of the standard and OPIRA implementations of SIFT, SURF and Ferns were evaluated for the MagicLand marker image sequences. The results are shown in Figures 7.46-7.48. The OPIRA implementation had the highest number of feature matches and lowest MAE for these evaluations and was shown to provide the lowest error and highest percentage of successfully registered frames for all markers and video sequences in Section 6.2. For this reason, the OPIRA implementations of the SIFT, SURF and Ferns registration algorithms were used for the evaluation of the histogram equalisation across the other markers.

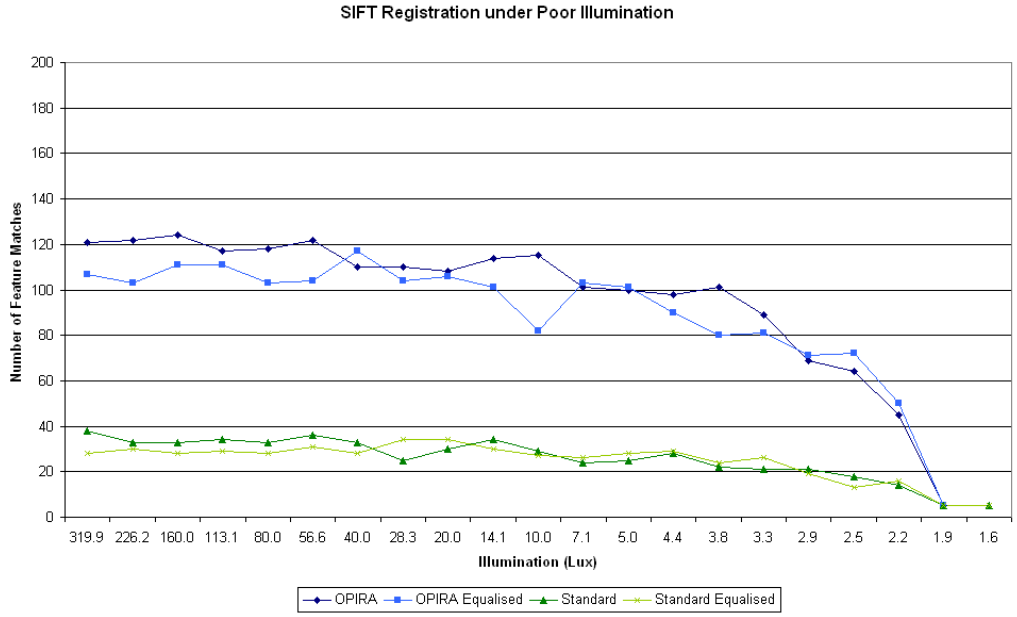
SIFT Table 7.37 shows the MAE of the SIFT algorithm for each marker before and after histogram equalisation. For three of the markers there was an improvement in the MAE, but the other five showed a decrease in accuracy. This is because the SIFT algorithm is already very good at illumination invariance, and by increasing the variance of illumination noise is amplified as well, which can cause erroneous registrations.

These results are consistent with the average number of feature matches found by SIFT shown in Table 7.38. While some markers benefited greatly from histogram equalisation such as the MacMini marker, the number of feature matches of other markers, such as the Stop marker, did not increase much.

Table 7.39 shows the percentage of successfully registered frames found by the SIFT algorithm for each marker before and after histogram equalisation.

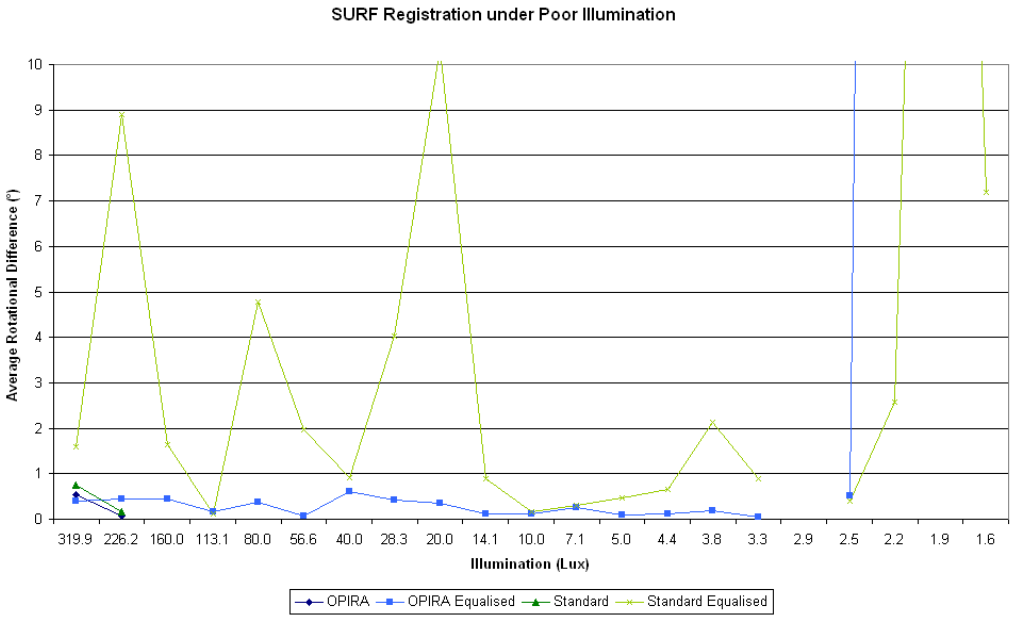


(a)

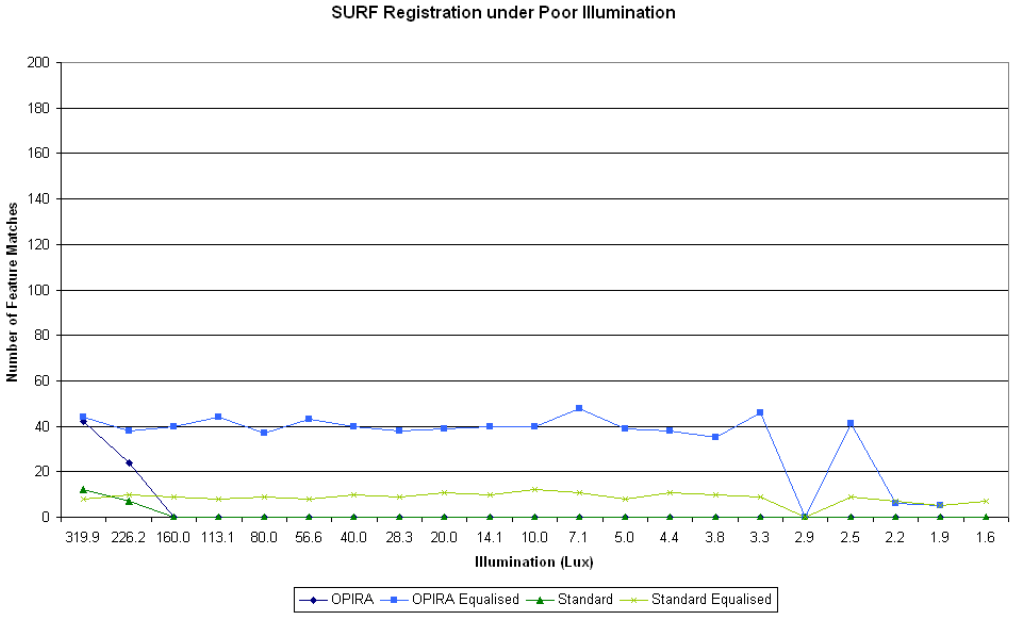


(b)

Figure 7.46: The average rotational difference (a) and number of feature matches (b) for the SIFT algorithm under different illuminations

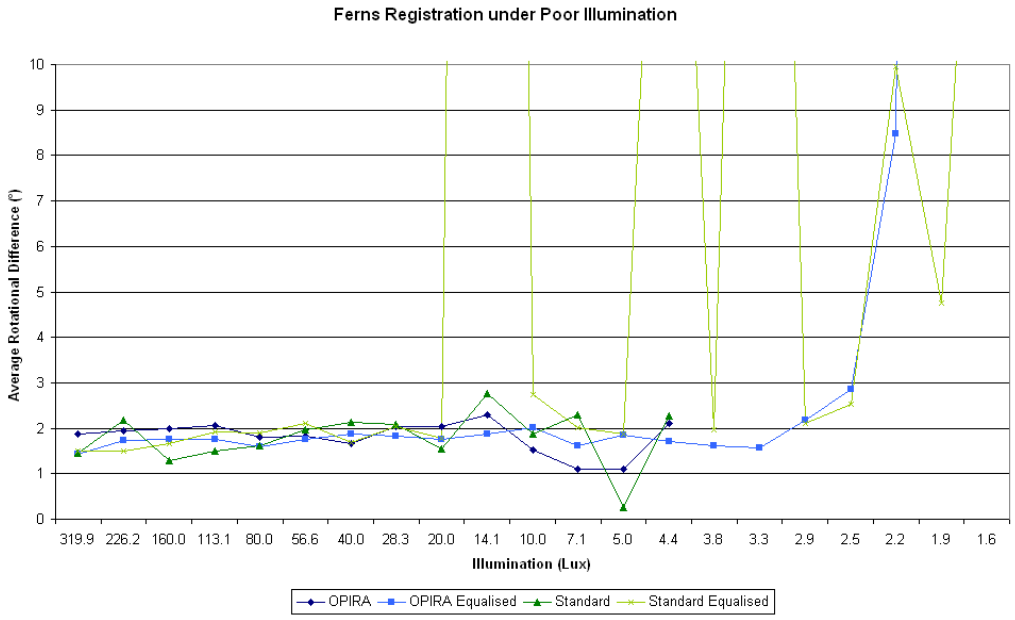


(a)

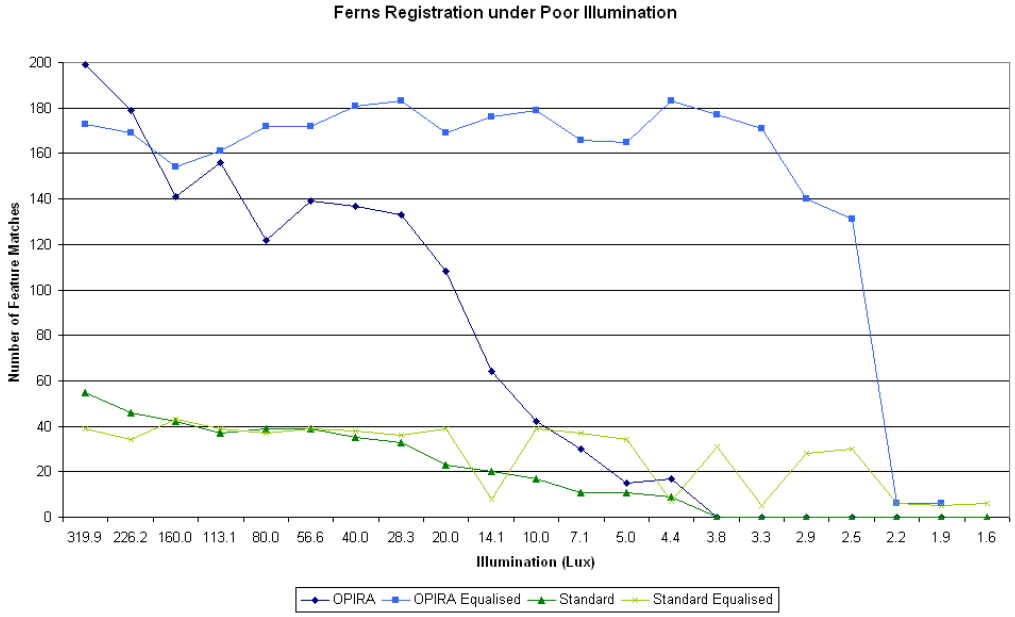


(b)

Figure 7.47: The average rotational difference (a) and number of feature matches (b) for the SURF algorithm under different illuminations



(a)



(b)

Figure 7.48: The average rotational difference (a) and number of feature matches (b) for the Ferns algorithm under different illuminations

The MacMini, Isetta and Grass markers saw an improvement, while the other markers had similar or worse results after histogram equalisation had been performed.

SURF Table 7.40 shows the MAE of the SURF algorithm for each marker before and after histogram equalisation. Only two of the markers had a decrease in the MAE, however 7.42 shows that most of the markers failed to have any successfully registered frames with an error of less than 5°. The increased MAE is due to the increased number of frames which were able to be registered by SURF after histogram equalisation had been performed. Those markers with lower MAE before histogram equalisation simply failed to register once the brightness had decreased a small amount.

The average number of feature matches shown in Table 7.41 correlates with this data. With the exception of the Grass marker, which SURF is unable to register due to its high level of repetitive detail, the average number of feature matches increased significantly after histogram equalisation was applied.

Table 7.42 shows the percentage of successfully registered frames found by the SIFT algorithm for each marker before and after histogram equalisation. By performing equalisation, the SURF algorithm was able to significantly increase the percentage of successfully registered frames for all Markers with the exception of the Grass marker.

Ferns Table 7.43 shows the MAE of the Ferns classifier for each marker before and after histogram equalisation. Five of the markers had a decrease in the MAE after histogram equalisation, although the MacMini saw a significant increase due to increased noise causing erroneous feature matches.

Table 7.44 shows that with histogram equalisation the average number of features matches found by the Ferns classifier increased for all markers.

With histogram equalisation, the percentage of successfully registered frames, shown in Table 7.45, increased for all markers using the Ferns classifier, with the exception of the MacMini marker, where increased noise caused erroneous matches and invalid registration calculation.

	Original	Equalised
MagicLand	4.15	4.52
Stop	130.53	139.42
Lucent	72.78	65.50
MacMini	110.47	58.28
Isetta	97.07	116.62
Philadelphia	13.92	19.85
Grass	219.24	166.51
Wall	23.69	41.71

Table 7.37: Mean absolute error of the SIFT algorithm for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	89	82
Stop	28	21
Lucent	68	87
MacMini	57	224
Isetta	33	34
Philadelphia	140	141
Grass	4	19
Wall	133	112

Table 7.38: Average number of feature matches found by the SIFT algorithm for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	86%	86%
Stop	5%	0%
Lucent	55%	55%
MacMini	23%	50%
Isetta	18%	27%
Philadelphia	77%	68%
Grass	0%	9%
Wall	68%	73%

Table 7.39: Percentage of successfully registered frames by the SIFT algorithm for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	0.31	7.36
Stop	74.74	98.13
Lucent	118.05	78.45
MacMini	144.24	37.70
Isetta	20.00	40.67
Philadelphia	12.63	22.08
Grass	∞	∞
Wall	3.56	25.88

Table 7.40: Mean absolute error of the SURF algorithm for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	3	33
Stop	11	34
Lucent	6	99
MacMini	6	199
Isetta	23	62
Philadelphia	136	194
Grass	0	0
Wall	60	145

Table 7.41: Average number of feature matches found by the SURF algorithm for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	9%	82%
Stop	0%	0%
Lucent	0%	59%
MacMini	0%	68%
Isetta	5%	14%
Philadelphia	41%	73%
Grass	0%	0%
Wall	18%	68%

Table 7.42: Percentage of successfully registered frames by the SURF algorithm for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	1.82	14.26
Stop	81.52	81.39
Lucent	79.68	64.65
MacMini	59.88	146.84
Isetta	97.10	27.09
Philadelphia	47.45	25.70
Grass	141.05	146.85
Wall	70.24	38.28

Table 7.43: Mean absolute error of the Ferns classifier for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	67	139
Stop	31	54
Lucent	118	124
MacMini	177	124
Isetta	50	105
Philadelphia	128	173
Grass	41	91
Wall	112	151

Table 7.44: Average number of feature matches found by the Ferns classifier for each marker before and after histogram equalisation

	Original	Equalised
MagicLand	64%	82%
Stop	0%	5%
Lucent	41%	59%
MacMini	50%	45%
Isetta	5%	27%
Philadelphia	55%	82%
Grass	18%	36%
Wall	41%	68%

Table 7.45: Percentage of successfully registered frames by the Ferns classifier for each marker before and after histogram equalisation

Overall results The SIFT registration algorithm did not benefit significantly from the use of the Histogram Equalisation. Although the MAE decreased and percentage of successfully registered frames increased for some of the markers, these improvements were minor in most cases, and registration of other markers was less accurate after Histogram Equalisation. The SIFT algorithm is extremely robust to changes in illumination, and the need for equalisation is unnecessary in most cases.

In contrast to SIFT, the SURF registration algorithm improved significantly with Histogram Equalisation. Although the MAE of some markers was lower before equalisation, this is because registration failed completely after the illumination was decreased slightly. The percentage of successfully registered frames increased significantly for almost all the markers after histogram equalisation, with the MacMini marker increasing 68% and the Lucent marker increasing 59%. The SURF algorithm improved a significant amount using equalisation, this is likely due to reduced illumination invariance from the approximations made to increase the speed of the algorithm.

The Ferns classifier also showed improvements using histogram equalisation. Five of the eight markers had a lower MAE, and seven markers had an increased percentage of successfully registered frames. While the improvements were not as significant as those seen with the SURF algorithm, the Ferns classifier can still benefit from the use of equalisation.

7.5 Marker Sources

As discussed in Section 6.3, the source of the marker used for registration can affect the accuracy of feature matching. A highly detailed source image will provide more detail and a greater number of features, but these matches may not correlate well to the images captured by the camera if are they are of lower quality. A lower detailed source which has been captured from the camera will have a higher feature similarity for the feature matching stage, but will be affected by any noise present when the image is captured.

To measure the effect of the source and quality of the marker image on the, six variations of each marker were created. The high quality digital image was digitally reduced in size to create three markers with vertical resolutions

(a) (b) (c)

(d) (e) (f)

Figure 7.49: The MagicLand markers, (a-c) Digitally created, (d-f) Captured by the camera.

of 160, 320 and 640. The high quality source image was printed in black and white onto an A4 sheet of paper with a RICOH Aficio MP 2550 printer at 200dpi. An image of this marker was captured using the camera described in Section 7.1.1 at 1280×960 resolution. This image was digitally resized to create markers with vertical resolutions of 160, 320 and 640.

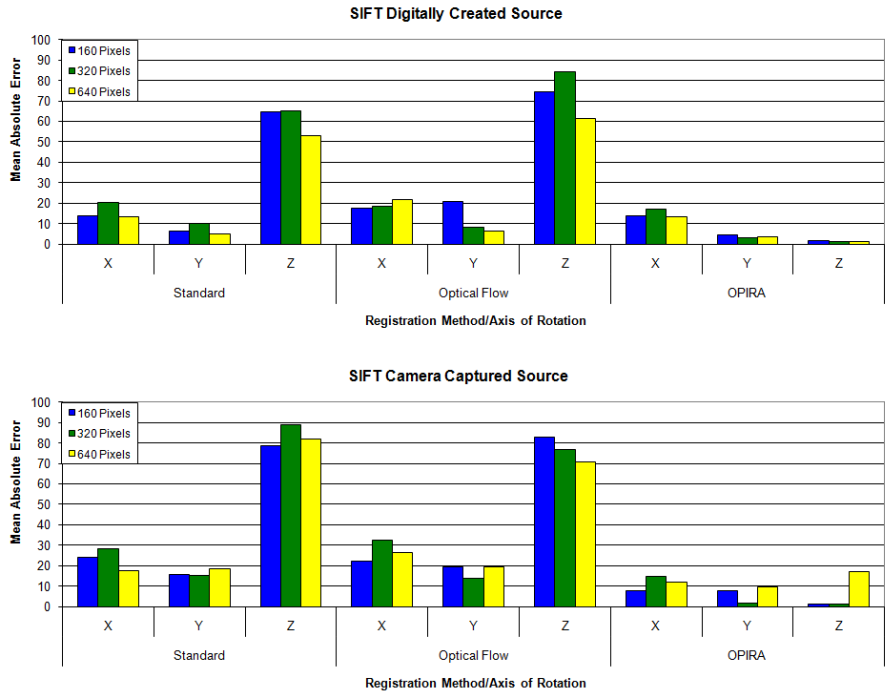
The images of the MagicLand marker used for this evaluation are shown in Figure 7.49. The markers are identified by the source, either Digital or Camera Capture, and the vertical resolution.

The performance of the standard, optical flow and OPIRA implementations of SIFT, SURF and Ferns were evaluated for the MagicLand marker image sequences. Registration was performed on the three axis rotation sequences using standard, optical flow and OPIRA implementations of the SIFT, SURF and Ferns registration algorithms for each of the marker conditions. The MAE and average number of feature matches for each evaluation are shown in the Figures 7.50-7.52. The result for registration of the 160 resolution markers is shown in blue, 320 resolution in red and 640 resolution in yellow.

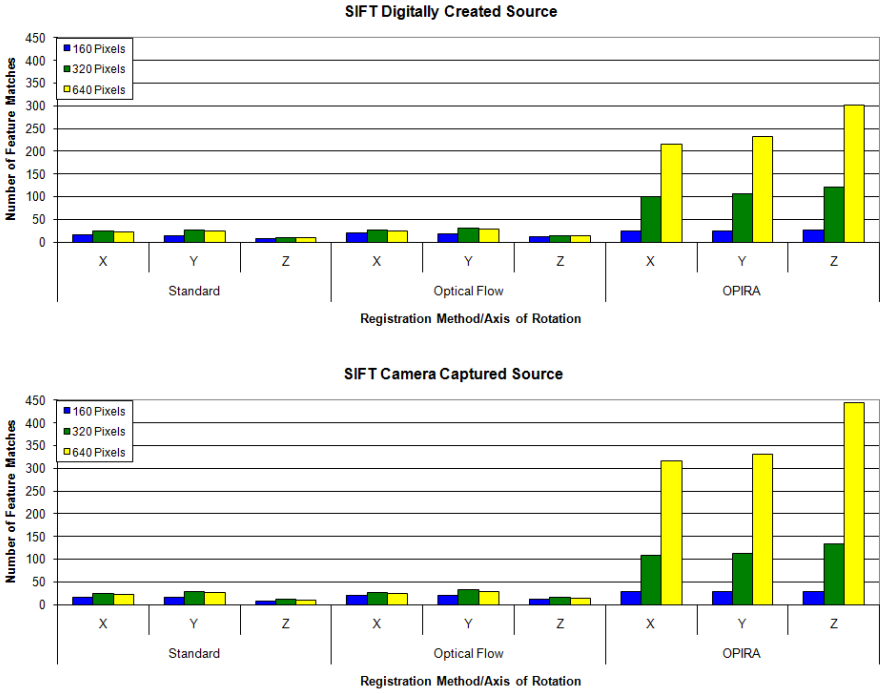
The OPIRA implementation had the highest number of feature matches and lowest MAE for these evaluations and was shown to provide the lowest error and highest percentage of successfully registered frames for all markers and video sequences in Section 6.2. For this reason, the OPIRA implementations of the SIFT, SURF and Ferns registration algorithms were used for the evaluation of the marker source and scale across the other markers.

SIFT Table 7.46 shows the MAE of the SIFT algorithm for each marker source and scale. The digitally created markers had lower error for the 320 resolution, while the camera captured markers had lower error for the smaller 160 resolution markers. Lower resolutions than the optimal lacked detail, while higher resolutions had too many features which were not visible in the rotational sequences which lead to erroneous registrations. The SIFT algorithm had the lowest overall error at 320 resolution digitally created markers.

Table 7.47 shows the average number of feature matches found by the SIFT algorithm for each marker source and scale. For both digitally created

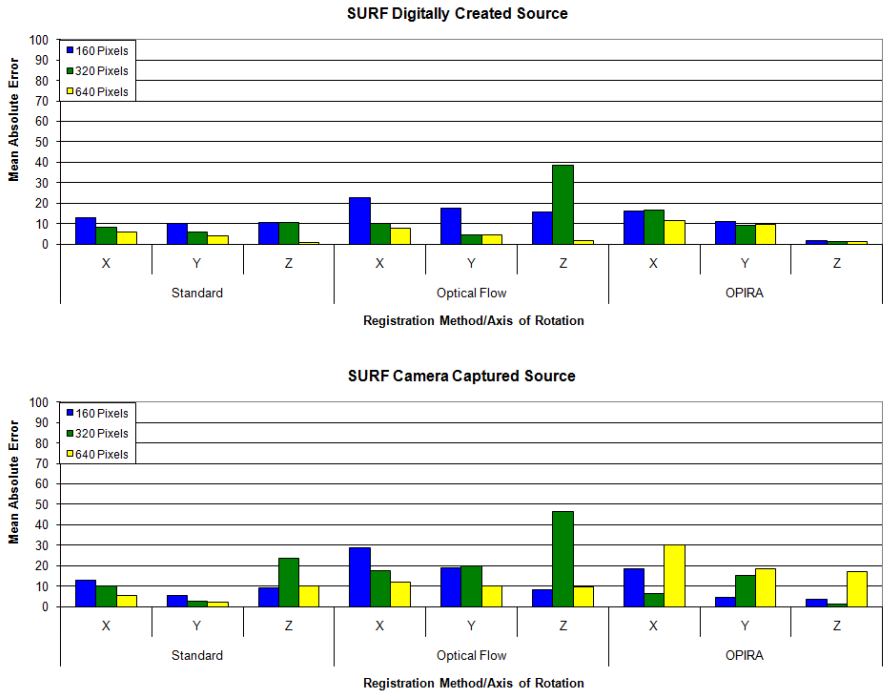


(a) Mean Absolute Error

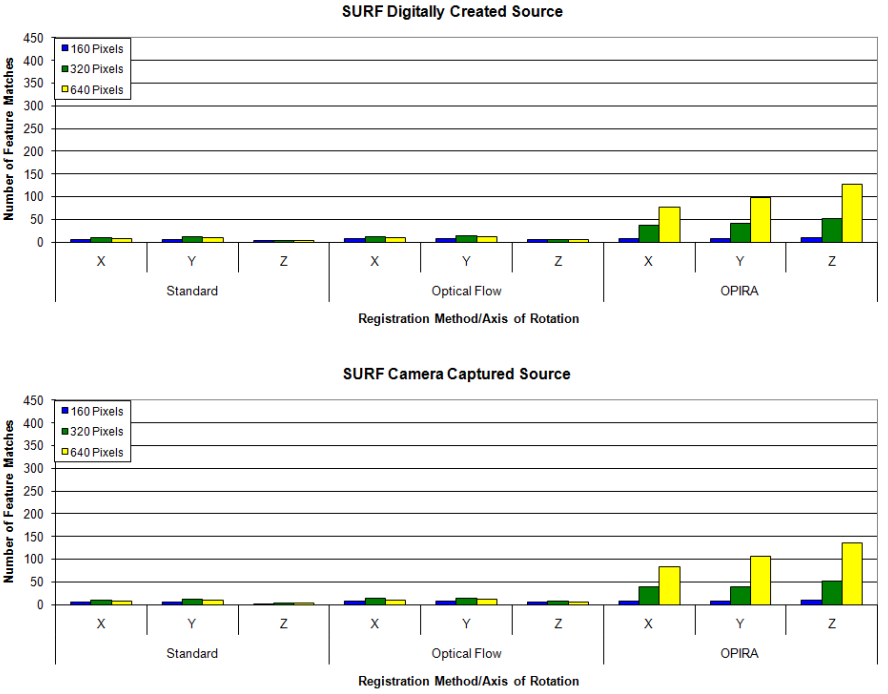


(b) Feature Matches

Figure 7.50: The SIFT algorithm with different marker sources and resolutions for the MagicLand marker, (a) The mean absolute error, (b) Number of Feature Matches

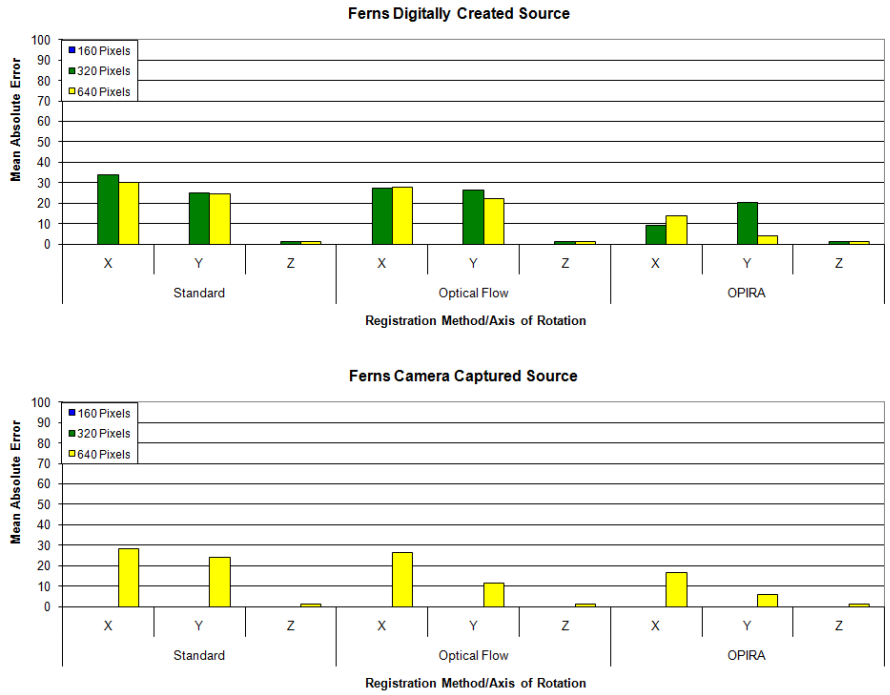


(a) Mean Absolute Error

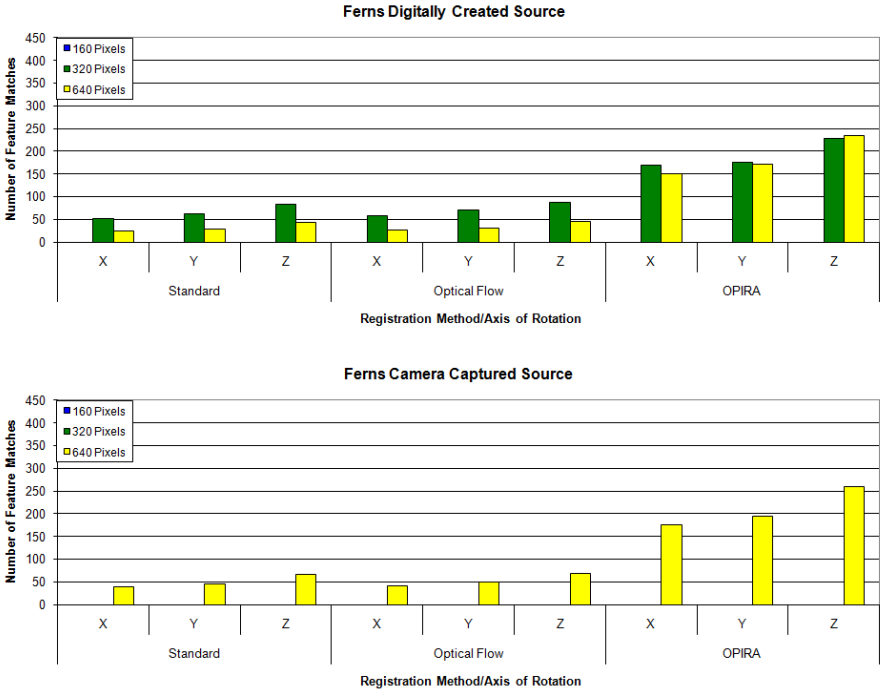


(b) Feature Matches

Figure 7.51: The SURF algorithm with different marker sources and resolutions for the MagicLand marker, (a) The mean absolute error, (b) Number of Feature Matches



(a) Mean Absolute Error



(b) Feature Matches

Figure 7.52: The Ferns classifier with different marker sources and resolutions for the MagicLand marker, (a) The mean absolute error, (b) Number of Feature Matches

and camera captured markers, increasing the size of the marker resulted in more feature matches, however at higher resolutions some of these features matches were erroneous, leading to increased error as seen in Table 7.46. The camera captured markers had a higher number of feature matches on average compared with the digitally created markers.

Table 7.48 shows the percentage of frames the SIFT algorithm was successfully able to register. For both digitally created and camera captured markers, the 320 resolution had the highest percentage of successfully registered frames for the rotational video sequences used. The SIFT algorithm had highest percentage of successfully registered frames at 320 resolution, though the difference between digitally created and camera captured markers was not significant.

SURF Table 7.49 shows the MAE of the SURF algorithm for each marker source and scale. For both digitally created and camera captured markers, the 640 resolution had the lowest overall error for the rotational video sequences used. The SURF algorithm had lower MAE with the camera captured markers, likely due to the closer similarity between features in the marker and video images. The SURF algorithm had the lowest overall error at 640 resolution camera captured markers.

Table 7.50 shows the average number of feature matches found by the SURF algorithm for each marker source and scale. For both digitally created and camera captured markers, increasing the size of the marker resulted in more feature matches. The camera captured markers had a higher number of feature matches on average compared with the digitally created markers.

Table 7.51 shows the percentage of frames the SURF algorithm was successfully able to register. The SURF algorithm had highest percentage of successfully registered frames at 640 resolution, though the difference between digitally created and camera captured markers was not significant.

Ferns Table 7.52 shows the MAE of the Ferns classifier for each marker source and scale. For both digitally created and camera captured markers, there was a total failure to register the 160 resolution markers. The resolution with the lowest MAE error depended on the marker. The digitally created

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	6.67	7.13	5.94	5.46	5.86	12.92
Stop	19.52	17.31	29.76	18.29	21.32	20.29
Lucent	23.61	10.31	25.45	14.18	17.23	16.79
MacMini	14.61	5.74	7.52	15.47	13.86	14.78
Isetta	15.37	17.54	22.72	12.54	14.75	17.80
Philadelphia	8.79	9.24	16.02	6.55	11.74	17.18
Grass	57.94	12.08	12.93	25.06	12.75	15.46
Wall	13.88	10.58	21.29	11.21	13.51	16.58

Table 7.46: Mean Absolute Error of the SIFT algorithm for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	26	109	249	28	118	363
Stop	28	37	33	30	48	65
Lucent	20	100	244	38	170	457
MacMini	27	238	547	41	294	744
Isetta	24	50	53	35	81	155
Philadelphia	44	154	260	64	213	450
Grass	4	50	257	11	108	509
Wall	26	136	321	44	227	587

Table 7.47: Average number of Feature Matches found by the SIFT algorithm for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	79%	77%	76%	78%	82%	78%
Stop	73%	71%	59%	72%	74%	72%
Lucent	61%	69%	68%	63%	70%	71%
MacMini	69%	82%	79%	59%	77%	80%
Isetta	74%	79%	66%	77%	82%	74%
Philadelphia	86%	87%	77%	85%	87%	81%
Grass	8%	70%	69%	55%	76%	72%
Wall	82%	87%	81%	77%	84%	82%

Table 7.48: Percentage of successfully registered frames by the SIFT algorithm for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	9.55	9.03	7.52	8.85	7.51	21.81
Stop	49.82	56.19	45.92	82.95	38.78	29.78
Lucent	69.17	33.47	14.71	61.30	25.23	16.29
MacMini	22.15	13.35	16.24	27.92	17.34	7.85
Isetta	20.33	10.43	14.43	21.94	19.25	10.72
Philadelphia	11.29	15.12	7.98	11.31	10.69	7.85
Grass	∞	∞	68.82	∞	∞	∞
Wall	11.31	15.84	14.16	18.35	7.66	12.56

Table 7.49: Mean Absolute Error of the SURF algorithm for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	9	43	100	8	43	108
Stop	5	6	6	4	6	7
Lucent	5	24	83	5	32	120
MacMini	10	119	323	11	127	389
Isetta	14	30	45	15	34	53
Philadelphia	35	156	250	45	199	344
Grass	0	0	0	0	0	0
Wall	13	81	116	18	101	142

Table 7.50: Average number of Feature Matches found by the SURF algorithm for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	54%	72%	68%	61%	76%	71%
Stop	10%	21%	22%	2%	28%	39%
Lucent	3%	43%	65%	8%	49%	57%
MacMini	43%	62%	71%	49%	63%	71%
Isetta	57%	70%	70%	63%	70%	77%
Philadelphia	75%	80%	88%	72%	81%	86%
Grass	0%	0%	0%	0%	0%	0%
Wall	59%	72%	76%	52%	75%	79%

Table 7.51: Percentage of successfully registered frames by the SURF algorithm for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	∞	10.31	6.19	∞	∞	7.85
Stop	∞	23.16	21.62	∞	17.85	18.95
Lucent	∞	16.55	28.57	∞	22.38	21.14
MacMini	∞	12.45	24.66	∞	21.43	35.34
Isetta	∞	22.56	11.18	∞	19.88	14.73
Philadelphia	∞	11.25	18.63	∞	10.04	17.66
Grass	∞	26.91	76.51	∞	26.90	37.73
Wall	∞	16.97	16.91	∞	15.37	13.98

Table 7.52: Mean Absolute Error of the Ferns classifier for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	0	191	185	0	0	210
Stop	0	41	38	0	43	45
Lucent	0	155	74	0	172	159
MacMini	0	216	131	0	215	144
Isetta	0	85	89	0	93	112
Philadelphia	0	171	113	0	210	177
Grass	0	65	7	0	83	72
Wall	0	155	123	0	195	172

Table 7.53: Average number of Feature Matches found by the Ferns classifier for each marker source and resolution

	Digitally created			Camera captured		
	160	320	640	160	320	640
MagicLand	0%	73%	75%	0%	0%	80%
Stop	0%	71%	71%	0%	72%	73%
Lucent	0%	71%	59%	0%	70%	69%
MacMini	0%	81%	70%	0%	76%	57%
Isetta	0%	76%	80%	0%	78%	81%
Philadelphia	0%	85%	80%	0%	88%	83%
Grass	0%	65%	12%	0%	65%	58%
Wall	0%	84%	79%	0%	84%	83%

Table 7.54: Percentage of successfully registered frames by the Ferns classifier for each marker source and resolution

markers had a lower average error than the camera captured markers.

Table 7.53 shows the average number of feature matches found by the SIFT algorithm for each marker source and scale. For both digitally created and camera captured markers, the Ferns classifier on average found a greater number of features matches using 320 resolution markers. The camera captured markers had a higher number of feature matches on average compared with the digitally created markers.

Table 7.54 shows the percentage of frames the Ferns classifier was successfully able to register. For both digitally created and camera captured markers, the Ferns classifier on average had a higher percentage of successfully registered frames using 320 resolution markers. The difference between digitally created and camera captured markers was not significant.

Overall results The SIFT registration algorithm had the lowest error and highest percentage of successfully registered frames when using the 320 resolution digitally created markers. The 160 resolution markers did not have enough detail, while the 640 resolution markers had more detail than could be seen in the video sequences and did not give better performance. The digitally created markers performed better overall than the camera captured markers, although in many cases the difference was minimal.

In contrast to SIFT, the SURF registration algorithm had the lowest overall error and highest percentage of successfully registered frames at the 640 resolution. The reason for this is likely due to SURF having a less distinctive feature descriptor than SIFT. The camera captured markers had a lower MAE when using SURF, although the difference between digitally created markers and camera captured markers had little effect on percentage of successfully registered frames.

The Ferns classifier was completely unable to register any markers at 160 resolution. The optimal resolution for low MAE with the Ferns classifier depended on the marker used, although the number of feature matches and percentage of successfully registered frames was on average higher using the 320 resolution markers. Digitally created markers have a lower average error but also lower average number of feature matches with the Ferns classifier, but there was no significant difference between the marker sources with re-

spect to the percentage of successfully registered frames.

7.6 Summary

In this chapter, the design and implementation of a uniform evaluation environment and software framework was discussed. A testing framework utilising an external inertial orientation sensor as a ground truth was used to evaluate the solutions proposed in Chapters 5 and 6 to reduce the negative impact of common image transformations and deformations. These evaluations included the OPIRA method for changes in scale, rotation, and perspective distortion, the improvements of the Wiener filter on blur, and the improvements of histogram equalisation on poor illumination. Additionally, the effect of marker source on feature matching was examined.

The OPIRA method for natural feature registration was found to significantly improve perspective and rotation invariance for all evaluated natural feature registration algorithms for every measurement. The success of the Wiener filter and histogram equalisation depended on the level of image distortion and the registration algorithm used. The optimal source and resolution of markers differed depending on which registration algorithm was used.

In the following chapter, the results of the evaluations are discussed. Observations are made about the effectiveness and feasibility of each solution in the context of natural feature registration.

Chapter 8

Discussion of Results

In this chapter the results of the evaluations conducted in the previous chapter are discussed. Each improvement is evaluated for its effectiveness, with the feasibility of implementation in a natural feature registration application examined.

8.1 OPIRA

OPIRA is a new method of natural feature registration presented in this research which improves invariance to changes in scale, rotation and perspective. Three evaluations were conducted to analyse the improvements possible when using OPIRA for registration.

8.1.1 Visual Inspection

The first evaluation, described in Section 7.2.1 was a quantitative assessment of OPIRA using visual inspection. Registration was performed on rotational sequences using standard, optical flow and OPIRA implementations of the SIFT and SURF registration algorithms. Each registration was given a rating of 0, 1 or 2 based on the accuracy of alignment of a virtual rectangle rendered on the marker.

The OPIRA implementation of the SIFT algorithm showed significant improvements across all rotational sequences. In the X axis rotation sequence, the failure limit of the standard and optical flow implementations was 60°, while the OPIRA implementation failed at 90°. In the Y axis rotation sequence, the failure limit of the standard implementation was 10°, the optical flow implementation failed at 40°, and the OPIRA implementation failed at 80°. In the Z axis rotation sequence, the failure limit of the standard implementation was 20°, the optical flow implementation failed at 30°, and full 360° rotation invariance was achieved by the OPIRA implementation.

Like the SIFT algorithm, the OPIRA implementation of the SURF algorithm showed significant improvements across all rotational sequences. In the X axis rotation sequence, the failure limit of the standard implementation was 70°, the optical flow implementation failed at 80°, and the OPIRA implementation failed at 90°. In the Y axis rotation sequence, the failure limit of the standard implementation was 60°, the optical flow implementation failed at 70°, and the OPIRA implementation failed at 85°. In the Z axis rotation sequence, the failure limit of the standard implementation was 30°, the optical flow implementation failed at 40°, and full 360° rotation invariance was achieved by the OPIRA implementation.

8.1.2 Perspective Invariance

The second evaluation of OPIRA assessed the standard, optical flow and OPIRA implementations of the SIFT, SURF and Ferns algorithms against the inertial ground truth for 8 different markers, as described in Section 7.2.2. The implementations were evaluated on X and Y rotational video sequences described in Section 7.1.5.

The OPIRA implementation of all tested algorithms on average had a lower mean absolute error (MAE) for each marker type tested than the standard or optical flow methods of registration. In the cases when the MAE of OPIRA was higher, this was either due to the standard and optical flow implementations completely failing to register most frames, or highly repetitive textures confusing the SURF registration algorithm.

The average number of feature matches found by the OPIRA implementation was on average two times higher than the optical flow and standard implementations for SURF, and three or more times higher for SIFT and Ferns for all markers. The only instance where OPIRA did not have a significantly higher average number of feature matches was the low textured Stop sign using the SURF algorithm, as SURF was unable to locate any more additional features.

The results for the percentage of successfully registered frames for this evaluation correlated with the number of feature matches. The OPIRA implementation had a higher percentage than the standard and optical flow

implementations for SIFT, SURF and Ferns for all markers. The only instance where there was no improvement was with the low textured Stop sign using the SURF algorithm.

The OPIRA implementation significantly improved the perspective invariance of the SIFT, SURF and Ferns algorithms for both rotation around the X and Y axes for all metrics used. The percentage of successfully registered frames increased by 15% for SIFT, 25% for SURF and 20% for Ferns.

8.1.3 Rotation Invariance

Section 7.2.3 assessed the rotational invariance of OPIRA in comparison with the standard and optical flow implementations. The implementations were evaluated on the Z rotation video sequence described in Section 7.1.5. Rotation dependent implementations of the SIFT and SURF algorithms were compared against their rotation independent implementations, but as the Ferns classifier is inherently rotationally invariant it was not be evaluated.

The OPIRA implementation of the rotation dependent SIFT and SURF algorithms had a significantly lower mean absolute error than the standard and optical flow implementations, with the exception of the low detailed Stop marker under the SURF algorithm. The OPIRA implementations of the rotation independent SIFT and SURF performed as good as or better the standard and optical flow implementations, and although these algorithms are meant to be rotation independent, the OPIRA implementation's rotation invariance had a significantly lower MAE for the Grass marker for SIFT and the Lucent marker for SURF.

Although there was not a significant difference in MAE between the standard and OPIRA implementations of the rotationally independent SIFT and SURF algorithms, the average number of feature matches found by OPIRA was at least three times greater for most markers. This difference was even greater when using the rotationally dependent SIFT and SURF algorithms, as the standard implementation found even fewer feature matches on average.

The OPIRA implementations of both the rotation dependent and rotation independent SIFT and SURF algorithms were able to successfully register almost 100% of frames. Across all markers and with both algorithm, this

result was as good as or better than the standard implementation of the rotational independent algorithms, and significantly better than the rotation dependent implementations.

Overall, OPIRA provided rotation invariance which was as good as, or better than, the rotational independent implementations of the SIFT and SURF algorithms, and was able to make rotational dependent implementations almost 100% rotationally invariant.

8.1.4 Selection Process

The evaluations in the previous chapter show OPIRA improves the perspective and rotation invariance of planar natural feature registration regardless of the algorithm or marker. These improvements come at the cost of computational efficiency. For each frame of registration OPIRA calculates registration of the original frame, optical flow, and registration of the rectified image.

A solution to this issue is the Fast-OPIRA registration implementation, described in Section 5.3.1, which only performs the computations necessary to ensure a robust registration. The feasibility of Fast-OPIRA was evaluated in Section 7.2.4 by examining which of the three methods was selected by the voting process at each frame during a video sequence involving rotation around all three axes.

As shown in Figure 7.34, the majority of the homography calculations are done using optical flow, with occasional registration of the rectified image to increase the number of features tracked. Registration of the original image is only performed when OPIRA initialises. These results prove that the Fast-OPIRA approach can improve the computational speed of OPIRA, while maintaining the same level of robustness to changes in scale, rotation and perspective distortion.

8.2 Blur Invariance

As discussed in Section 6.1, blur is detrimental to the accuracy of registration as it degrades the quality of the images captured by the camera. Out-of-focus and motion blur are two common forms of blur, and the method of removal

for both is identical and the results for removal of one blur type will correlate to the results for the other. In this research, both blur types were evaluated by removal of out-of-focus blur, as this type of blur can be more accurately emulated.

The evaluation, described in Section 7.3.2, assessed the improvements possible in registration using the Wiener filter to reduce out-of-focus blur. Each markers three rotation sequences were blurred at five discrete levels using point spread functions obtained from the defocused camera. Registration using the OPIRA implementation of the SIFT, SURF and Ferns algorithms were conducted on these blurred sequences before and after the Wiener filter had been applied.

There was an increase in MAE for all algorithms as the scale of blur increased. When the Wiener filter was applied, this increase in MAE with scale was still present, but the rate of increase was far less. The SIFT and Ferns algorithms have a lower MAE for all markers after filtering, and the SURF algorithm, which failed after the third level of blur before filtering for almost all markers, was able to continue to register up to the fifth level of blur after filtering.

As the level of blur increased, the average number of feature matches for all markers decreased. As seen in the MAE results, by applying the Wiener filter, this rate of decline is reduced after using the Wiener filter, for the SIFT, SURF and Ferns algorithms.

The percentage of successfully registered frames is higher after Wiener filtering for almost all markers and levels of blur. The SIFT and Ferns algorithm had over 60% registration success for almost all markers after the fourth level of blur using the filter, while without it all markers dropped below 50% after the third level of blur. The SURF algorithm had 50% success after the fourth level with the filter, while the blurred images dropped below 50% after the second level of blur.

The use of the Wiener filter to reduce the problem of blur is worthwhile when the level of blur is large enough to reduce registration accuracy. All the registration algorithms evaluated benefited from the Wiener filter. Blur is not an invertible operation, as the level of data corruption increases the improvements gained from the use of the Wiener filter decrease. The results

of this evaluation show an increase in the mean absolute error and a decrease in the number of feature matches and percentage of successfully registered frames as the blur level increases. The Wiener filter reduces the rate of degradation of registration accuracy, but eventually the amount of blur will be too great for the filter to improve registration.

8.3 Illumination Invariance

In Section 6.2, histogram equalisation was proposed as a method to improve robustness to poor illumination. To evaluate this, images of each marker were captured in an environment with precisely controlled lighting as the light was decreased from a typical office lighting level of 320 lux to 0 lux. The evaluation is described in Section 7.4.

The MAE of the SIFT and SURF algorithm increased after histogram equalisation for most markers, but for very different reasons. The SIFT algorithm is already extremely robust to poor illumination, and the increased level of noise due to histogram equalisation caused erroneous feature matches which caused poor registration. In contrast, the SURF algorithm failed to register completely with equalisation after a minimal reduction in illumination, while the accuracy of registration for the equalised images degraded slowly, resulting in a higher error. The Ferns classifier showed on average lower MAE with histogram equalisation.

The average number of feature matches found increased for the SURF and Ferns algorithms for almost all markers. However, the SIFT algorithm had similar or fewer feature matches after histogram equalisation, due to erroneous feature matches caused by the increased noise level reducing the accuracy of homography computation and valid matches being removed by RANSAC.

The percentage of successfully registered frames increased considerably for both the SURF and Ferns algorithms with the use of histogram equalisation for almost all markers. Although the SIFT algorithm had an increase in the percentage of successfully registered frames for a few of the markers, the rest had a similar or decreased percentage as before equalisation was performed.

The improvements gained by using histogram equalisation to reduce the problem of poor illumination depend on the registration algorithm. The SIFT algorithm is highly robust to poor illumination, and in many cases the increased noise after histogram equalisation caused a decrease in performance after histogram equalisation. The SURF algorithm is highly affected by changes in illumination, and registration can improve considerably with histogram equalisation at low light levels. The Ferns algorithm also improved after histogram equalisation.

Poor illumination is not an invertible operation, as illumination and the signal to noise ratio decrease, so do the improvements gained from histogram equalisation. Histogram equalisation reduces the rate of degradation of registration accuracy, but eventually a threshold will be reached where the equalisation is unable to improve registration.

8.4 Marker Sources

The improvements evaluated in Sections 7.2-6.1 were designed to remove detrimental effects of common image transformations and distortions during the feature detection and description stages of natural feature registration. The accuracy of the feature matching stage depends on the correlation between features in the marker and frame.

To evaluate the effect of different marker sources and resolutions on feature matching, a high quality, digitally created and low quality, camera captured source of each marker were collected. Each source was digitally resized to generate three resolutions identified by their vertical resolutions: 160, 320, and 640 pixels. The three rotational video sequences of each marker were registered using the OPIRA implementations of the SIFT, SURF and Ferns algorithms trained with the six generated markers.

The SIFT algorithm had the lowest MAE at 320 resolution for the digitally created markers and 160 resolution for the camera captured sources. The lowest overall error across all markers for SIFT was 320 resolution digitally created markers. The SURF algorithm had the lowest MAE at 640 resolution for both the digitally created and camera captured markers, with the lowest overall error being the 640 resolution camera captured markers.

The Ferns Classifier failed to register any of the 160 resolution markers, and there was no clear optimal resolution. The digitally created markers had lower average error than the camera captured markers for Ferns.

The average number of feature matches increased as the size of the marker increased for the SIFT and SURF algorithms, while Ferns had the highest average at 320 resolution. For all algorithms, the camera captured markers had a higher average number of feature matches than the digitally created markers, although in the case of SIFT and Ferns, many of these feature matches were erroneous.

The SIFT and Ferns algorithms had the highest percentage of successfully registered frames with the 320 resolution images for both the digitally created and camera captured sources, and the difference between the sources was not significant. The SURF algorithm had the highest percentage of successfully registered frames at 640 resolution, with no significant difference between the sources.

The exact reason that registration performs better on the 320 resolution marker instead of the 640 resolution marker for SIFT and Ferns is likely due to the detail level visible in the marker in the evaluation video sequences. It is expected the results would be different with a better quality camera and a higher quality print of the marker. With a six degree of freedom ground truth, it is expected a zooming sequence would show greater accuracy using the higher resolution images. This is something which is planned in future, as discussed in Section 9.2.3.

The camera captured markers had a higher number of feature matches than the digitally created markers for every registration algorithm. This increased number of matches is likely due to the increased correlation between features identified in the marker and features identified in the frame. In a digitally created marker, the properties intrinsic to the camera such as lens distortion are not present, and the features identified in the marker will not have these properties. However, in the case of SIFT and Ferns, these matches were often erroneous and increase the MAE.

Overall, the 320 resolution digitally created markers were optimal for SIFT and Ferns with the highest percentage of successfully registered frames and lowest MAE. For SURF, the 640 resolution camera captured markers

had the highest percentage of successfully registered and lowest MAE.

8.5 Summary

This chapter discussed the results of the evaluations conducted in Chapter 7. The effectiveness of each proposed solution was presented, and the feasibility of use in a natural feature registration algorithm was discussed.

OPIRA was shown to increase the percentage of successfully registered frames under perspective invariance of SIFT, SURF and Ferns by 15%, 25% and 20% respectively. The rotation dependent SIFT and SURF algorithms were almost 100% rotationally invariant with the use of the OPIRA implementation. The OPIRA implementation of the rotation depended SIFT and SURF algorithms had a lower error than the rotation invariant algorithms.

The Wiener filter increased the limits of natural feature registration under blur for all algorithms. The more blurred an image is, the greater the improvements, although this is only true until a threshold is reached where the image becomes too degraded to restore.

Histogram equalisation showed considerable improvements for the SURF and Ferns algorithms, although the highly illumination invariant SIFT algorithm actually had worse results due to the increased noise present.

320 Resolution digitally created markers had the highest accuracy for the SIFT and Ferns algorithms, while the SURF algorithm performed best using the 640 resolution camera captured markers.

In the following chapter, applications which have used this research are presented, and future directions of the research are discussed.

Chapter 9

Applications and Future Work

This chapter discusses some of the applications using the outcomes of this research, as well as the future research directions.

9.1 Applications

The framework developed as an outcome of this research, especially the OPIRA implementation for natural feature registration algorithm, has been integrated into a number of applications. In this section some of the significant applications are presented.

9.1.1 OSGART

OSGART is a "software development framework ... for Rapid Application Development in the domain of Mixed Reality" (Looser, Grasset, Seichter and Billinghurst 2006). OSGART combines OpenSceneGraph (OSG)[1], a powerful open source three dimensional graphics toolkit, with the ARToolKit marker based registration library to facilitate creation of complex augmented reality applications.

In collaboration with Julian Looser (HIT Lab NZ), the natural feature registration framework was integrated into OSGART to create rich augmented reality applications using natural feature markers. Figure 9.1 shows several examples of high resolution augmented reality using the OPIRA implementation of the SURF algorithm.

9.1.2 Jack the Time Traveller MagicBook

The natural feature registration OSGART library has been used to create a new augmented reality book, or "MagicBook", for the Australian Cen-

[1] `http://www.openscenegraph.org/`

Figure 9.1: Examples of high resolution 3D content overlaid on natural markers using OSGART and the OPIRA implementation of SURF

tre for the Moving Image[2], developed at the Human Interface Technology Laboratory New Zealand (HIT Lab NZ).

The previous MagicBook, "Giant Jimmy Jones" (McKenzie and Darnell 2003) used the ARToolKit NFT (Kato et al. 2003) platform, which required an ARToolKit fiducial marker on each page to initialise the natural feature registration. The presence of the fiducial marker was visually distracting from the book's artwork, and was a powerful motivation for the use of the OPIRA implementation of natural feature registration.

This new MagicBook, titled "Jack the Time Traveller", is a comic book which tells the story of a man named Jack as he travels through time helping to ensure past events run smoothly. Figure 9.2 shows the four marker pages with the augmented reality scenes.

Figure 9.2: Pages of "Jack the Time Traveller" with augmented content overlaid

[2] http://www.acmi.net.au/

To guarantee a high level of interactivity, specific modifications were made to the OPIRA implementation. As every second page in the book is a natural feature marker, only a single marker can be visible at a time. Because of this, feature matching did not have to be conducted on every marker in the complete marker set for each frame, but only on the previous successfully registered marker. Feature matching of the complete marker set was only performed when the number of matched features dropped below a threshold.

The Fast-OPIRA implementation was used for the MagicBook to increase the registration speed. Registration of the original frame was only performed when the number of feature matches fell below a threshold, experimentally found to be optimal at 10 matches. Registration of the rectified image and optical flow were performed every frame for the last recognised marker.

As described in Section 5.3.1, Fast-OPIRA supports the use of separate registration algorithms for registration of the original image and registration of the rectified image. For the MagicBook, slow rotational invariant SURF was performed on the original image to locate markers regardless of the orientation, and fast rotational dependent SURF was used for the rectified image, with rotation invariance provided by OPIRA. This increased the speed of registration by approximately six times.

This MagicBook had four different scenes, each one requiring a separate marker. The markers were generated from a digital copy of the book, and resized to 250×353 resolution. By working with the artists, the pages were designed to contain ideal features such as visible corners and points, with gradient colouring surrounding them.

To maximise the frame rate while maintaining a high quality render, video was captured at 640×480 resolution and used for rendering, and then digitally resized to 320×240 resolution for the registration. These speed improvements allowed the MagicBook software to operate at over 20 frames per second with four markers on a standard desktop computer.

The new MagicBook was displayed at the central library of the University of Canterbury, New Zealand[3], as shown in Figure 9.3. The book was also featured at the Australian Centre for the Moving Image for the "Screen

[3] `http://www.hitlabnz.org/wiki/3D_Magicbook_display_at_UC_library`

Worlds: The story of Film, Television and Digital Culture" Exhibition[4].

Figure 9.3: The "Jack the Time Traveller" kiosk in the University of Canterbury central library

9.1.3 Esperient Creator

The Esperient Creator[5] is a three dimensional content authoring tool designed to allow intuitive creation of standalone or web based content with high visual content. In collaboration with Julian Looser (HIT Lab NZ), a plug-in was developed for the Esperient Creator, which supports and integrates the natural feature registration framework. This solution offers a unique tool for designing and developing augmented reality applications based on natural feature registration that can be deployed as standalone or web applications.

[4] `http://www.acmi.net.au/screen_worlds.htm`

[5] `http://www.esperient.com/`

Figure 9.4 shows a screen capture of an augmented reality advertisement displayed on a product box, running the OPIRA implementation of SURF within the Esperient Creator software.

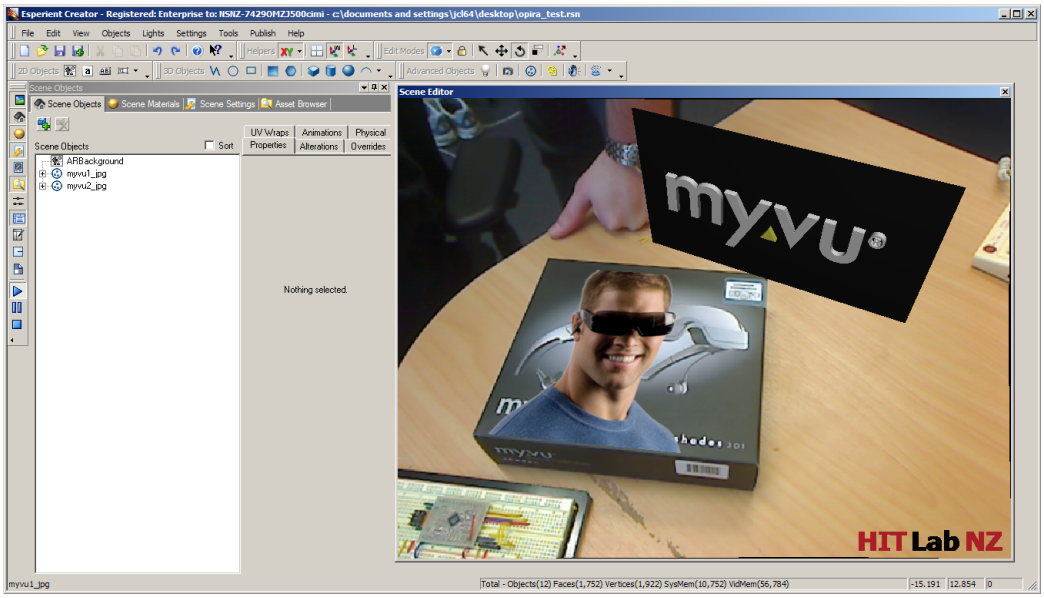


Figure 9.4: The OPIRA library running with SURF registration in the Esperient Creator client

9.1.4 OPIRA Robotics Platform

The previous applications use the registration framework as the enabling computer vision technology for augmented reality. This framework has also been combined with a low cost robotics platform using the parallel port on a computer to control a radio controlled car. Video is sent wirelessly from a camera mounted to the car to the computer via a receiver attached to a USB video capture device. This system, shown in Figure 9.5, allows the computer to automatically control the car using video obtained from the car's perspective.

The costs of the platform can be kept low by processing video on the computer instead of relying on an embedded processor on the car. This has

Figure 9.5: The low cost computer vision driven robotics platform

the added advantage of the high processing power of desktop machines compared to embedded processors. Many cheap robots like this can be networked together to provide a robotic "swarm", which has applications in navigation and mapping (Pack and Mullins 2003).

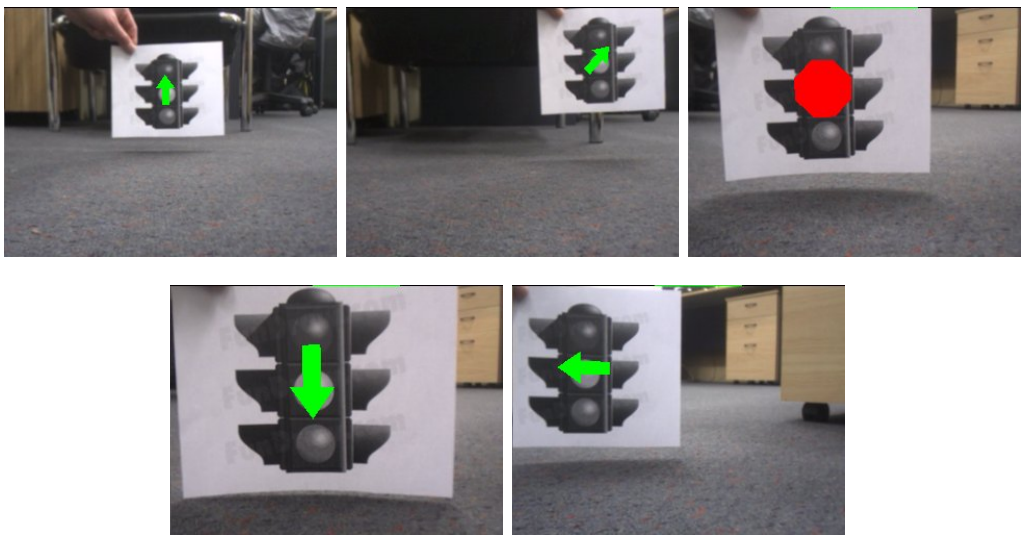


Figure 9.6: The view from the perspective of the robot as it follows a marker, the arrow indicates the direction of travel.

A simple example application was written which has the car follow a marker, as shown in the sequence of images in Figure 9.6. The car drives forwards if the marker is beyond a certain distance threshold from the camera, and backwards if the marker is closer than another distance threshold. The car turns left or right if the marker is beyond defined thresholds in the X axis.

9.2 Future Work

The follow sections discuss future research directions for the work presented in this thesis.

9.2.1 OPIRA optimisations

The main contribution of this thesis is the OPIRA implementation of registration. OPIRA has shown to significantly improve the invariance of leading natural feature registration algorithms to changes in rotation and perspective. In this section, further improvements are proposed for the OPIRA implementation.

Despite their limitations, discussed in Chapter 2, fiducial registration algorithms have an advantage over natural feature registrations in their ability to register a large number of markers in a single application.

In the new MagicBook described in Section 9.1.2, four markers were used in the same augmented reality application. In order to perform registration in real time, OPIRA was modified to only search for the last seen marker. An application may require tens or even hundreds of markers to be registered within the same application.

A proposed method of dealing with a large database of markers is the use of the bag of words algorithm (Lewis 1998); a method of information retrieval using a spare data set. The bag of words algorithm has been successfully used to identify objects in an image using natural features (Sivic, Russell, Efros, Zisserman and Freeman 2005), and could be used to quickly determine markers visible within a scene before the costly process of registration.

Another proposed improvement to OPIRA is illumination equalisation and mapping. OPIRA rectifies the marker within an image to remove the

effects of scale, rotation and perspective. A model of the illumination in the scene could be obtained from image patches between the rectified image and the original marker. This model could be used to remove undesirable illumination effects, or even to provide lighting information to augmented reality to create a more realistic scene (Chen, Yang, Xiao and Ding 2008).

Finally, further improvements to the speed of OPIRA are planned. Fast-OPIRA increases the speed of OPIRA significantly by favouring tracking over registration, reducing the average speed of transformation computation to less than normal natural feature registration algorithms. However, when tracking accuracy drops and registration is required, there is a reduction in the frame rate. A multi-threaded version of the library is being developed to take advantage of the increased prevalence of multi core machines. This multi-threaded system predicts tracking failure and performs registration in a separate thread, such that when tracking does fail, a new set of points are available, with no noticeable decrease in speed. It is also capable of registering each marker in separate thread, increasing the speed of multi-marker applications.

9.2.2 Adaptive Filtering and Registration System

In Chapters 5 and 6, solutions were presented which improve the invariance of natural feature registration to common image transformations and deformations. In particular, histogram equalisation was employed to reduce the negative effect of poor illumination, and the Wiener filter reduced the problem of global noise. However, as discussed in Chapter 8, in certain circumstances these two solutions can degrade registration accuracy. In addition to this, although not explored in this research, these solutions have computational overhead.

For an application which requires real-time performance, the increase in computational load required for these solutions may not be worth the benefits they provide. For example, Section 7.3.2 shows that for low levels of blur, the improvements gained using the computationally expensive Wiener filter are slight, and in fact sometimes the accuracy of registration after the filter was applied was worse than before it was applied.

These principles also apply to natural feature registration algorithms. Depending on the task, if registration robustness is less important than speed, SURF may be the optimal registration algorithm, while if the contrary is true, the SIFT registration algorithm may be more suitable.

In applications where the user may not be familiar with the different natural feature registration algorithms and image filters available, such as authoring tools like the Esperient Creator (discussed in 9.1.3), it is desirable to have the framework automatically select the appropriate registration algorithm for the task. The proposed solution for this is an adaptive artificial neural network system, which determines the optimal combination of filters and registration algorithms (Clark and Green 2005).

Properties of the marker and operating environment, called image metrics, are identified by the system and used by the artificial neural network framework to establish the optimal registration framework. Examples of possible image metrics are shown in Figure 9.7.

Relationships between metrics, filters and registration algorithms are complex. An artificial neural network can be trained to model complex relationships, and is very suitable for this sort of application. Users could add additional registration algorithms and filters to the system and, with adequate training, the accuracy of the system will only ever improve.

Figure 9.8 shows the pipeline proposed for the artificial neural network framework. Image metrics are extracted from the marker and environment after a few calibration frames. These metrics, in conjunction with the training data, are supplied to the artificial neural network. The neural network chooses the optimal registration algorithm based on the relationships established in the training.

The environmental metrics, registration algorithms chosen, and additional training data are supplied to another neural network to determine and initialise optimal filters for the registration algorithm given the operating environment. The registration algorithm parameters are calibrated using environmental metrics obtained from the video, and the registration framework is established.

The establishment of this framework is part of the calibration stage of registration, and does not affect the speed of registration once calibration

Figure 9.7: A selection of properties which can be obtained from an image (Left to Right, Top to Bottom): Original image, Connected components, Optical flow vectors, Intensity, Spectral components, Hue/Saturation, High frequency components, Areas of similar intensity, Difference between frames, Motion point spread function, Eigen values

has concluded. Optionally, the artificial neural network system can be used to generate a new pipeline if the accuracy of registration falls below a desired threshold due to changing environmental conditions.

A preliminary development of this framework and pilot evaluation of this concept was conducted using ARToolKit NFT (Kato et al. 2003) and SIFT (Lowe 2004) registration algorithms on the original MagicLand marker using a pre-recorded image sequence. Even with limited training data, the artificial neural network system registered 45% more frames than ARToolKit NFT, and 130% more frames than the SIFT algorithm.

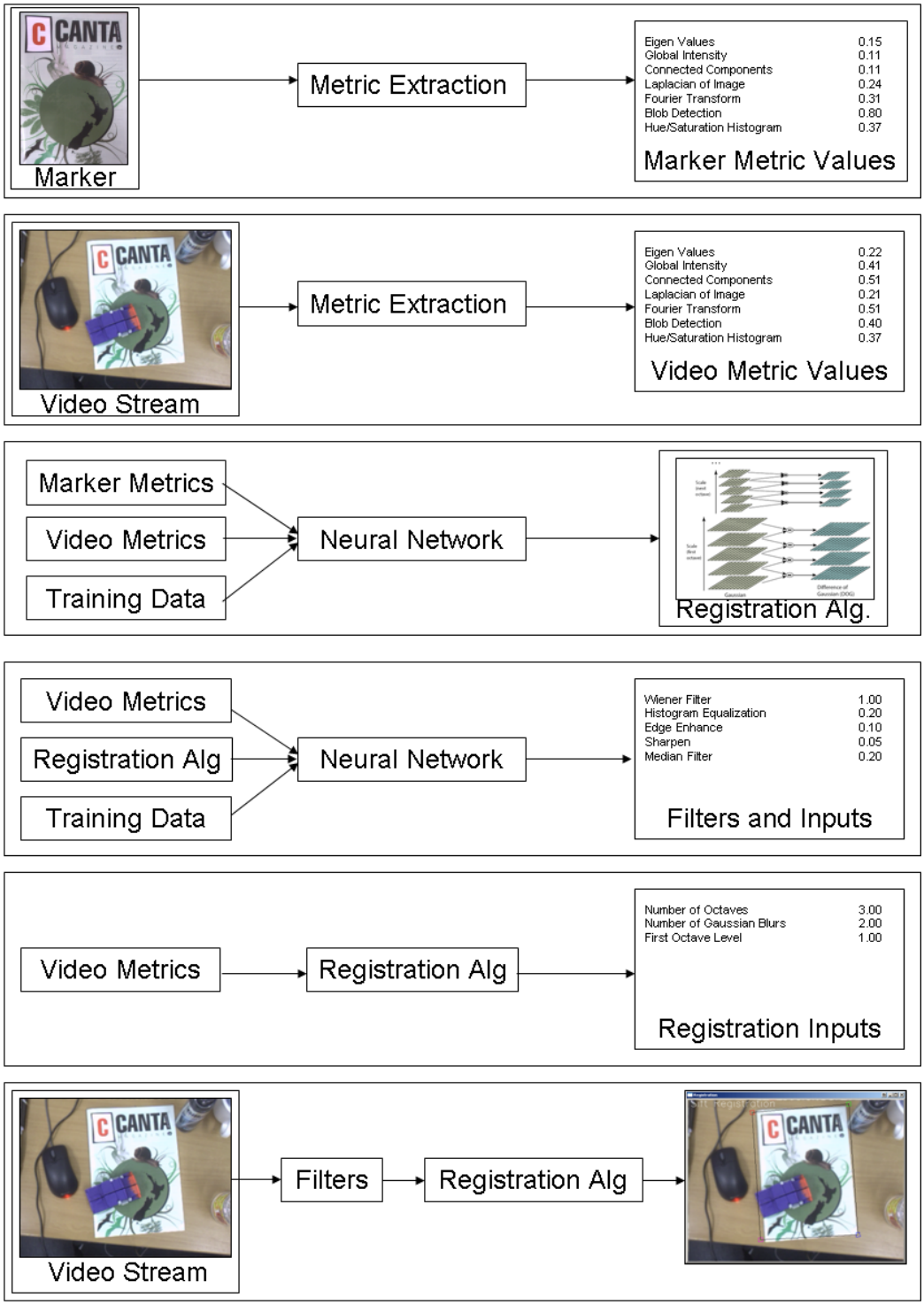


Figure 9.8: A suggested pipeline approach using an artificial neural network to determine optimal filter and registration algorithm combinations

9.2.3 Six Degree of Freedom Ground Truth

The solutions proposed in this research were evaluated against a three dimensional inertial orientation sensor ground truth, as described in Section 7.1.5. The sensor provided a highly accurate measurement of the rotational accuracy of registration, but was unable to evaluate the accuracy in translation. An evaluation of translation accuracy requires a six degree of freedom (6DOF) external ground truth.

A six degree of freedom ground truth can be modelled using a simulation of camera optics (Smit and van Liere 2008). Despite the increasing accuracy of these simulations (Klein and Murray 2008), it is the imperfections in the capture of images which often provide the greatest challenges to registration, and the only way to obtain realistic imperfections is to use a camera.

Preliminary work has been initiated in the area with a 6DOF ground truth system using a highly accurate optical tracking system, consisting of four ARTtrack2 IR-Cameras[6]. The DTrack software[7] provided with the cameras allows computation of the position and orientation of rigid bodies with retro-reflective markers. By attaching rigid bodies to the marker and the camera, the system can represent the two objects with six degrees of freedom.

A registration matrix can be derived if the transformation between a marker and camera and the camera intrinsic parameters are known, as described in Chapter 3. In Figure 9.9, an image of a checker board pattern is shown from a camera (top), with the camera (grey cone) and marker rendered as found by the 6DOF ground truth (bottom). The three coordinate systems shown in the bottom of Figure 9.9 are the camera coordinate system, marker coordinate system, and coordinate system of the ground truth system.

In Figure 9.10, a wooden camera calibration platform from a previous project is shown with the 6DOF rigid bodies attached. The transformation between the camera and the marker can be found by the difference between the position and orientation of the rigid bodies. In this figure the camera and marker are fixed, however the 6DOF ground truth supports dynamic motion

[6] `http://www.ar-tracking.de/ARTtrack2.52+B6Jkw9.0.html`

[7] `http://www.ar-tracking.de/DTrack1.238+B6Jkw9.0.html`

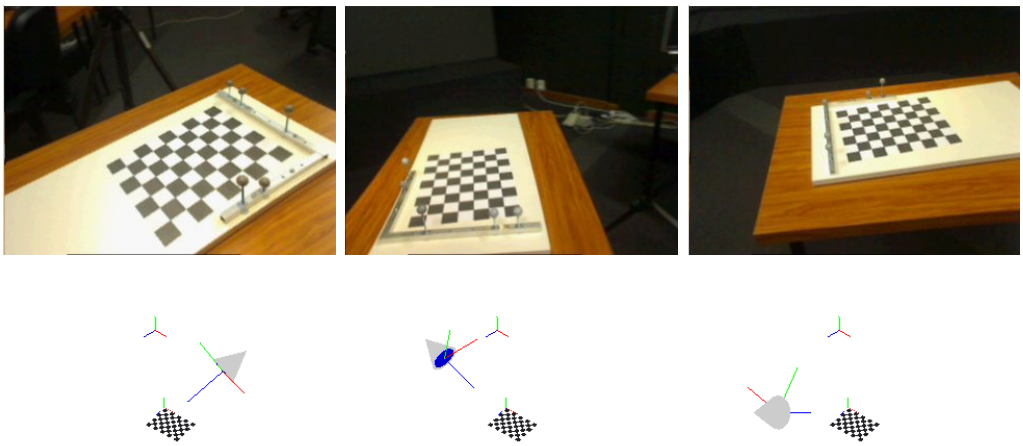


Figure 9.9: Images taken from the camera(top), and a visualisation of the positions and orientations of the camera and marker as found by the 6DOF ground truth (bottom)

capture of both objects.

With the new six degree of freedom ground truth, new measurements of registration accuracy can be used. For example, the difference between the computed feature positions and the actual feature positions can provide a measure of feature repeatability, and the accuracy of feature matching can be evaluated by evaluating the error of each match.

The data obtained from the 6DOF ground truth is perfectly suited to be training data for the adaptive artificial neural network system described in Section 9.2.2.

After development of the 6DOF ground truth is finished, there are plans to create a freely available library of video sequences with the ground truth data for other researchers to evaluate their own registration algorithms. The libraries will include environments featuring varied lighting, cameras, and markers. These evaluation libraries have proven popular for colour models, such as the Amsterdam Library of Object Images (Geusebroek, Burghouts and Smeulders 2005), and edge detection, such as the Berkeley Segmentation Benchmark (Martin, Fowlkes, Tal and Malik 2001).

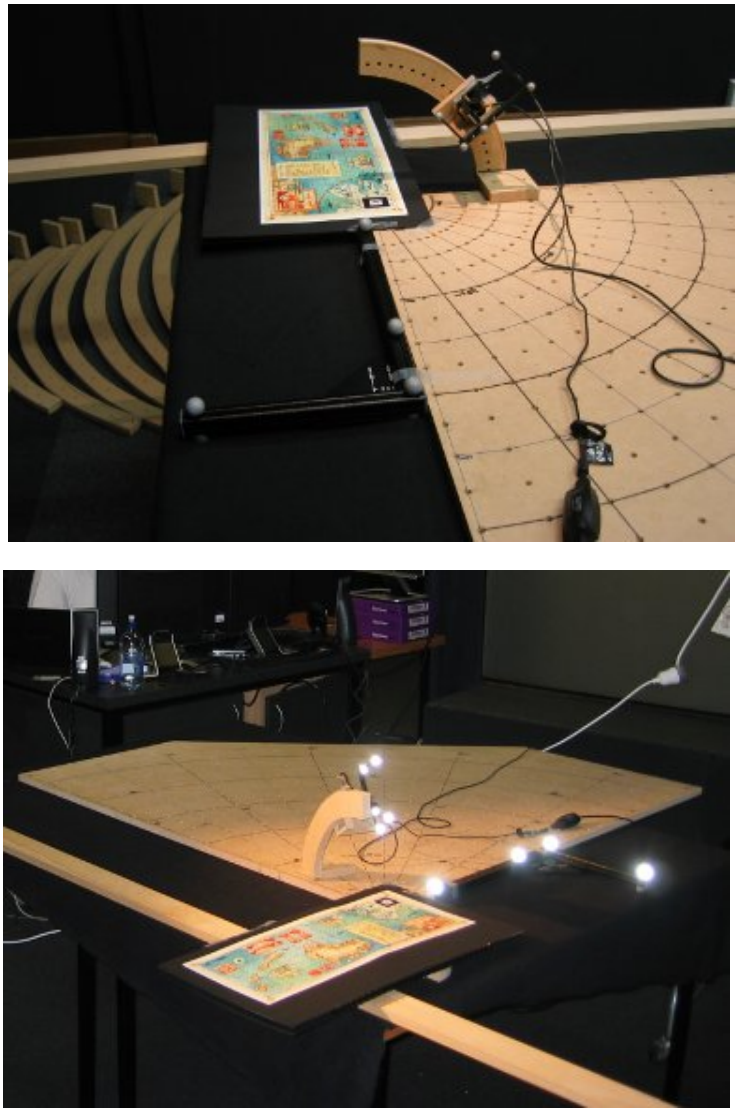


Figure 9.10: A large scale testing rig with the retro-reflective markers attached

9.3 Summary

In this chapter, some of the significant applications using the research in this thesis were presented. After this, future directions of the research were discussed.

The following chapter concludes this thesis with a concise summary of the research contributions contained in this thesis.

Chapter 10

Conclusion

10.1 Contributions

The main contributions of this thesis are as follows:

- A critical review of popular planar fiducial and natural feature registration algorithms.
- A comprehensive discussion of the theory of planar natural feature registration as a detailed analysis of the registration process.
- A study of the different steps of the registration process, leading to the proposal of methods of improvement and the introduction of a new algorithm OPIRA (Optical-flow Perspective Invariant Registration Augmentation).
- A detailed and systematic evaluation of the proposed improvements for natural feature registration algorithms, complemented by an analysis of the improvements of OPIRA. From these findings recommendations were proposed regarding development of natural feature registration systems.
- A proposed framework for minimising external environmental problems which affect the performance of natural feature registration.
- A software framework based around OPIRA for designing, testing and deploying natural feature registration algorithms. The framework is sufficiently robust to be used for augmented reality or robotic navigation.

10.2 Summary

Planar natural feature registration is the process of calculating the transformation between a camera and an object it is viewing, known as a marker. This transformation describes the position and orientation of the object with respect to the camera, and is essential for applications such as augmented reality, medical imaging, and robotic guidance.

The accuracy of a natural feature registration algorithm is reduced when the marker undergoes changes in scale, rotation and perspective with respect to the camera. Noise and poor illumination also degrade the effectiveness of natural feature registration. In this thesis, solutions to resolve these weaknesses were proposed and experimentally evaluated, and the effect of marker source and resolution on registration accuracy were also evaluated. A summary of the results of this series of evaluations is presented below.

- **OPIRA** Optical-flow Perspective Invariant Registration Augmentation (OPIRA) is a new method of registration that improves the registration robustness of natural feature registration algorithms with respect to changes in scale, rotation and perspective. The OPIRA implementation was shown to improve the percentage of successfully registered frames under perspective distortion for the SIFT by 15%, SURF by 25% and the Ferns Classifier by 20%.

  Full 360° rotation invariance was also achieved for the rotation dependent SIFT and SURF algorithms using the OPIRA implementation. The OPIRA implementation of these rotation dependent algorithms were as accurate or better than the rotation independent algorithms.

  OPIRA provides these improvements with no additional marker training time, with an increase of less than double in processing time on average. For applications where time is critical, a variant of OPIRA called Fast-OPIRA was proposed, which equals or exceeds the speed of leading natural registration algorithms with no observable reduction in robustness compared with the standard OPIRA implementation.

- **Noise Invariance** In Chapter 6, noise was classified into two classes,

local noise such as Gaussian and salt-and-pepper, and global noise such as blur. The feature description and matching stages of natural feature registration algorithms reduce the effects of local noise. For this reason, this research focused on removing global noise using the Wiener filter.

For all evaluated algorithms, the Wiener filter extended the invariance of natural feature registration algorithms in response to blur. Every algorithm had more accurate registration results at higher levels of blur, and SURF and Ferns were able to register at high levels of blur after the Wiener filter had been applied, when registration had failed completely when the Wiener filter was not used.

While the Wiener filter produces impressive results in the right circumstances, in certain cases the registration performance was degraded due to artefacts introduced during the filtering process. In addition, the Wiener filter is a computationally complex algorithm and can reduce the speed of registration. For these reasons, it is suggested that the Wiener filter is not a universal improvement for registration, but rather a solution to a specific problem.

- **Illumination Invariance** This research explores the use of histogram equalisation as a means of reducing the problem of poor illumination for natural feature registration. The results of the evaluations showed a significant improvement for the SURF and Ferns registration algorithms. The SIFT algorithm however showed no difference in the results of registration when using histogram equalisation.

  Like the Wiener filter, histogram equalisation is a solution to a specific problem, however as histogram equalisation is very computationally fast and not detrimental to registration, it can be applied universally.

- **Marker Source** The solutions in the preceding evaluations were designed to resolve the problems caused by image transformations and deformations in the feature detection and description stages of natural feature registration. To ensure accurate registration, the marker image the algorithm is trained to use must be highly correlated to the marker

as it appears to the camera.

Each marker was captured from two sources, a high quality digital source, and a low quality camera. Each source was resized to create three different resolutions. The digitally created markers had a high level of detail, but less correlation with the camera's perception of the marker, while the markers captured from the camera had the opposite properties.

The evaluations showed that there is no optimal combination of marker source and resolution for all natural feature registration algorithms. SIFT and Ferns worked best using the medium sized resolution digitally created markers, while SURF performed best using the largest size camera captured markers.

10.3 Future work

Chapter 9 discusses applications built using the results of this research, and proposed future directions. This future work is summarised here.

- **OPIRA optimisations** The OPIRA method of natural feature registration has shown to significantly improve leading natural feature registration algorithms. Future work plans to improve OPIRA further, in particular its ability to handle large numbers of markers, built in illumination mapping and invariance, and general speed improvements.

- **Adaptive Filtering and Registration System** The Wiener filter and histogram equalisation solutions to the problems of blur and poor illumination can provide great improvements to natural feature registration. However, these algorithms also increase the computational time required for registration, and in certain cases, can degrade the performance of registration. An artificial neural network system is proposed which can automatically choose the image filters and registration algorithms which will deliver optimal registration accuracy in an uncalibrated environment.

- **Six Degree of Freedom Ground Truth** In this research, a three degree of freedom ground truth was presented for evaluation of the proposed solutions. This ground truth provided an accurate measure of the angular accuracy of registration. A six degree of freedom ground truth is proposed in Chapter 9 which evaluates both the angular and translational accuracy of registration, with an unlimited range of motion. This ground truth will be used to generate data libraries that other researchers can evaluate their own natural feature registration algorithms against.

References

Azuma, R. and Bishop, G.: 1994, Improving static and dynamic registration in an optical see-through hmd, *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, pp. 197–204.

B., O. C., Xaio, F. and Middlin, P.: 2002, What is the best fiducial?, *The First IEEE International Augmented Reality Toolkit Workshop*, Darmstadt, Germany, pp. 98–105.

Bajura, M. and Neumann, U.: 1995, Dynamic registration correction in video-based augmented reality systems, *IEEE Comput. Graph. Appl.* **15**(5), 52–60.

Baker, S., Roth, S., Scharstein, D., Black, M. J., Lewis, J. and Szeliski, R.: 2007, A database and evaluation methodology for optical flow, *Computer Vision, IEEE International Conference on* **0**, 1–8.

Baumberg, A.: 2000, Reliable feature matching across widely separated views, *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on* **1**, 774–781 vol.1.

Bay, H., Tuytelaars, T. and Van Gool, L.: 2006, Surf: Speeded-up robust features, *9th European Conference on Computer Vision*, Graz, Austria.

Beis, J. S. and Lowe, D. G.: 1997, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, *In Proc. IEEE Conf. Comp. Vision Patt. Recog*, pp. 1000–1006.

Bellman, R.: 1961, *Adaptive control processes: a guided tour*, Princeton University Press, Princeton, N.J.,.

Bentley, J. L.: 1975, Multidimensional binary search trees used for associative searching, *Commun. ACM* **18**(9), 509–517.

Biemond, J., Lagendijk, R. and Mersereau, R.: 1990, Iterative methods for image deblurring, *Proceedings of the IEEE* **78**(5), 856–883.

Billinghurst, M., Cheok, A., Prince, S. and Kato, H.: 2002, Real world teleconferencing, *IEEE Computer Graphics and Applications* **22**(6), 11–13.

Billinghurst, M., Kato, H. and Poupyrev, I.: 2001, The magicbook - moving seamlessly between reality and virtuality, *Computer Graphics and Applications, IEEE* **21**(3), 6–8.

Bouguet, J. Y.: 2002, Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm.

Bradski, G.: 2000, The opencv library, *Dr. Dobb's Journal of Software Tools* .

Brown, M. and Lowe, D.: 2002, Invariant features from interest point groups, *In British Machine Vision Conference*, pp. 656–665.

Canny, J.: 1986, A computational approach to edge detection, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **8**(6), 679–698.

Chan, R. H., Ho, C. H. and Nikolova, M.: 2005, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization., *IEEE Transactions on Image Progressing* **14**(10), 1479–1485.

Chen, Y., Yang, X., Xiao, S. and Ding, X.: 2008, Relighting with real incident light source, *ISMAR '08: Proceedings of the 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, Washington, DC, USA, pp. 157–158.

Childers, D., Skinner, D. and Kemerait, R.: 1977, The cepstrum: A guide to processing, *Proceedings of the IEEE* **65**(10), 1428–1443.

Chinnasarn, K., Rangsanseri, Y. and Thitimajshima, P.: 1998, Removing salt-and-pepper noise in text/graphics images, *Circuits and Systems, 1998. IEEE APCCAS 1998. The 1998 IEEE Asia-Pacific Conference on*, pp. 459–462.

Cho, Y., Lee, J. and Neumann, U.: 1998, A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality, *Y. Cho, J. Lee, and U. Neumann, A Multi-ring Color Fiducial System and An Intensity-invariant Detection Method for Scalable Fiducial-Tracking Augmented Reality. In IWAR, 1998.* .

Cho, Y., Park, J. and Neumann, U.: 1997, Fast color fiducial detection and dynamic workspace extension in video see-through self-tracking augmented reality, *PG '97: Proceedings of the 5th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, p. 168.

Clark, A. and Green, R.: 2005, An adaptive algorithm switching system for image based object registration, *Image and Vision Computing New Zealand, 2005. IVCNZ 2005. 20th International Conference.*

Clark, A. and Green, R.: 2006, Detection and removal of global and local noise in realtime video streams, *Image and Vision Computing New Zealand, 2006. IVCNZ 2006. 21st International Conference*, pp. 367–372.

Clark, A., Green, R. and Grant, R.: 2007, Image and video noise - a comparison of noise in images and video with regards to detection and removal, *in* A. Ranchordas, H. Araújo and J. Vitrià (eds), *VISAPP 2007: Proceedings of the Second International Conference on Computer Vision Theory and Applications, Barcelona, Spain, March 8-11, 2007 - Volume 1*, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, pp. 153–156.

Clark, A., Green, R. and Grant, R.: 2008, Perspective correction for improved visual registration using natural features., *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pp. 1–6.

Edelman, S., Intrator, N. and Poggio, T.: 1997, Complex cells and object recognition, *Unpublished manuscript:* `http://kybele.psych.cornell.edu/~edelman/archive.html` .

Fabian, R. and Malah, D.: 1991, Robust identification of motion and out-of-focus blur parameters from blurred and noisy images, *CVGIP: Graph. Models Image Process.* **53**(5), 403–412.

Fiala, M.: 2004, Artag revision 1. a fiducial marker system using digital techniques, *Technical Report NRC 47419*, Institute for Informational Technology.

Fiala, M.: 2005a, Artag, a fiducial marker system using digital techniques, *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* **2**, 590–596 vol. 2.

Fiala, M.: 2005b, Comparing artag and artoolkit plus fiducial marker systems, *Haptic Audio Visual Environments and their Applications, 2005. IEEE International Workshop on*, pp. 147–152.

Fischler, M. A. and Bolles, R. C.: 1981, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* **24**(6), 381–395.

Fleet, D. J. and Weiss, Y.: 2005, *Mathematical Models in Computer Vision: The Handbook*, Springer, chapter 15, pp. 239–258.

Geusebroek, J. M., Burghouts, G. J. and Smeulders, A. W. M.: 2005, The amsterdam library of object images, *International Journal of Computer Vision* **61**(1), 103–112.

Grabner, M., Grabner, H. and Bischof, H.: 2006, Fast approximated sift, *ACCV 2006: Asian Conference on Computer Vision*, Springer-Verlag, pp. 918–927.

Grant, R., Green, R. and Clark, A.: 2007, Hue variance prediction - an empirical estimate of the variance within the hue of an image, *in* A. Ranchordas, H. Araújo and J. Vitrià (eds), *VISAPP 2007: Proceedings of the Second International Conference on Computer Vision Theory and Applications, Barcelona, Spain, March 8-11, 2007 - Volume 1*, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, pp. 5–9.

Grant, R., Green, R. and Clark, A.: 2008, HLS distorted colour model for enhanced colour image segmentation, *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pp. 1–6.

Hajnal, J. V., Hill, D. L. and Hawkes, D. J.: 2001, *Medical Image Registration (Biomedical Engineering)*, CRC Press Inc.

Harris, C. and Stephens, M.: 1988, A combined corner and edge detection, *Proceedings of The Fourth Alvey Vision Conference*, pp. 147–151.

Henrysson, A., Ollila, M. and Billinghurst, M.: 2005, Mobile phone based ar scene assembly, *MUM '05: Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, ACM, New York, NY, USA, pp. 95–102.

Horn, B. K. P. and Schunck, B. G.: 1981, Determining optical flow, *Artificial Intelligence* **17**, 185–203.

Indyk, P. and Motwani, R.: 1998, Approximate nearest neighbors: towards removing the curse of dimensionality, *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, ACM, New York, NY, USA, pp. 604–613.

Jin, F., Fieguth, P., Winger, L. and Jernigan, E.: 2003, Adaptive wiener filtering of noisy images and image sequences, *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Vol. 3, pp. III–349–52 vol.2.

Juan, M., Joele, D., Botella, C., Baños, R., Alcañiz, M. and van der Mast, C.: 2006, The use of a visible and/or an invisible marker augmented reality system for the treatment of phobia to small animals, *Annual Review of CyberTherapy and Telemedicine*, Vol. 4, pp. 33–38.

Jurie, F. and Dhome, M.: 2002, Hyperplane approximation for template matching, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(7), 996–1000.

Kato, H. and Billinghurst, M.: 1999, Marker tracking and hmd calibration for a video-based augmented reality conferencing system, *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on* pp. 85–94.

Kato, H., Tachibana, K., Billinghurst, M. and Grafe, M.: 2003, A registration method based on texture tracking using artoolkit, *Augmented Reality Toolkit Workshop, 2003. IEEE International*, pp. 77–85.

Ke, Y. and Sukthankar, R.: 2004, Pca-sift: a more distinctive representation for local image descriptors, *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* **2**, II–506–II–513 Vol.2.

Klein, G. and Murray, D.: 2008, Compositing for small cameras, *Proc. Seventh IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'08)*, Cambridge.

Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R. and Tuceryan, M.: 1997, Real-time vision-based camera tracking for augmented reality applications, *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM Press, Lausanne, Switzerland, pp. 87–94.

Lemuz-López, R. and Arias-Estrada, M.: 2006, Iterative closest sift formulation for robust feature matching, *Advances in Visual Computing, Second International Symposium, ISVC 2006 Lake Tahoe, NV, USA, November 6-8, 2006. Proceedings, Part II*, pp. 502–513.

Lepetit, V. and Fua, P.: 2005, Monocular model-based 3d tracking of rigid objects, *Found. Trends. Comput. Graph. Vis.* **1**(1), 1–89.

Lepetit, V. and Fua, P.: 2006, Keypoint recognition using randomized trees, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(9), 1465–1479.

Lepetit, V., Pilet, J. and Fua, P.: 2004, Point matching as a classification problem for fast and robust object pose estimation, *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Vol. 2, pp. II–244–II–250 Vol.2.

Lewis, D. D.: 1998, Naive (bayes) at forty: The independence assumption in information retrieval., *in* C. Nédellec and C. Rouveirol (eds), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Springer Verlag, Heidelberg, DE, Chemnitz, DE, pp. 4–15.

Li, G. and Song, B.: 2004, Image salt-pepper noise elimination by detecting edges and isolatednoise points, *in* A. C. Campilho and M. S. Kamel (eds), *Image Analysis and Recognition: International Conference, ICIAR 2004, Porto, Portugal, September 29-October 1, 2004, Proceedings, Part II*, Springer, pp. I: 171–178.

Lieberknecht, S., Benhimane, S., Meier, P. and Navab, N.: 2009, A dataset and evaluation methodology for template-based tracking algorithms, *ISMAR '09: Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, Washington, DC, USA, pp. 145–151.

Lindeberg, T.: 1994, Scale-space theory: A basic tool for analysing structures at different scales, *J. of Applied Statistics* **21(2)**, 224–270.

Lindeberg, T. and Garding, J.: 1997, Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure, *Image and Vision Computing* **15**, 415–434(20).

Looser, J., Grasset, R., Seichter, H. and Billinghurst, M.: 2006, Osgart - a pragmatic approach to mr, *International Symposium of Mixed and Augmented Reality (ISMAR 2006)*, ISMAR, Santa Barbara, CA, USA.

Lowe, D.: 1999, Object recognition from local scale-invariant features, *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Vol. 2, pp. 1150–1157 vol.2.

Lowe, D. G.: 2004, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**, 91–110.

Lucas, B. D. and Kanade, T.: 1981, An iterative image registration technique with an application to stereo vision (darpa), *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121–130.

Martin, D., Fowlkes, C., Tal, D. and Malik, J.: 2001, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *Proc. 8th Int'l Conf. Computer Vision*, Vol. 2, pp. 416–423.

Masso, L., Dhome, M. and Jurie, F.: 2003, Contour/texture approach for visual tracking, *Scandinavian Conference on Image Analysis*, pp. 661–668.

McKenzie, J. and Darnell, D.: 2003, The eyemagic book: A report into augmented reality storytelling in the context of a children's workshop, *Technical report*, Institute for Informational Technology.

Mikolajczyk, K. and Schmid, C.: 2002, An affine invariant interest point detector, *Proc. European Conf. Computer Vision*, Springer Verlag, pp. 128–142.

Mikolajczyk, K. and Schmid, C.: 2004, Scale & affine invariant interest point detectors, *Int. J. Comput. Vision* **60**(1), 63–86.

Mikolajczyk, K. and Schmid, C.: 2005, A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis & Machine Intelligence* **27**(10), 1615–1630.

Mohan, A., Woo, G., Hiura, S., Smithwick, Q. and Raskar, R.: 2009, Bokode: imperceptible visual tags for camera based interaction from a distance, *ACM Trans. Graph.* **28**(3), 1–8.

Moré, J. J.: 1978, The Levenberg-Marquardt algorithm: Implementation and theory, *Numerical Analysis*, Vol. 630 of *Lecture Notes in Mathematics*, Springer Berlin / Heidelberg, pp. 105–116.

Neumann, U. and Park, J.: 1998, Extendible object-centric tracking for augmented reality, *Virtual Reality Annual International Symposium, 1998. Proceedings IEEE 1998*, pp. 148–155.

Neumann, U. and You, S.: 1999, Natural feature tracking for augmented reality, *Multimedia, IEEE Transactions on* **1**(1), 53–64.

Ozuysal, M., Calonder, M., Lepetit, V. and Fua, P.: 2009, Fast keypoint recognition using random ferns, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* .

Ozuysal, M., Fua, P. and Lepetit, V.: 2007, Fast keypoint recognition in ten lines of code, *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* pp. 1–8.

Pack, D. and Mullins, B.: 2003, Toward finding an universal search algorithm for swarm robots, *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, Vol. 2, pp. 1945–1950.

Park, J., You, S. and Neumann, U.: 1998, Extending augmented reality with natural feature tracking, *in* M. R. Stein (ed.), *International Workshop on Augmented Reality (IWAR)'98*, Vol. 3524, SPIE, pp. 105–114.

Piekarski, W. and Thomas, B.: 2002, Using artoolkit for 3d hand position tracking in mobile outdoor environments, *Augmented Reality Toolkit, The First IEEE International Workshop*.

Pilet, J., Lepetit, V. and Fua, P.: 2005, Real-time non-rigid surface detection, *Conference on Computer Vision and Pattern Recognition, San Diego, CA*.

Rank, K., Lendl, M. and Unbehauen, R.: 1999, Estimation of image noise variance, *Vision, Image and Signal Processing, IEE Proceedings-* **146**(2), 80–84.

Rekimoto, J. and Ayatsuka, Y.: 2000, Cybercode: designing augmented reality environments with visual tags, *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, ACM, New York, NY, USA, pp. 1–10.

Ribo, M., Pinz, A. and Fuhrmann, A.: 2001, A new optical tracking system for virtual and augmented reality applications, *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, Vol. 3, pp. 1932–1936 vol.3.

Richardson, W. H.: 1972, Bayesian-based iterative method of image restoration, *J. Opt. Soc. Am.* **62**(1), 55–59.

Rosten, E. and Drummond, T.: 2005, Fusing points and lines for high performance tracking., *IEEE International Conference on Computer Vision*, Vol. 2, pp. 1508–1511.

Rosten, E., Porter, R. and Drummond, T.: 2008, Faster and better: A machine learning approach to corner detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **99**(1).

Russ, J. C.: 2002, *The Image Processing Handbook, Fourth Edition*, CRC Press Inc.

Schaffalitzky, F. and Zisserman, A.: 2002, Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?", *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, Springer-Verlag, London, UK, pp. 414–431.

Schmalstieg, D. and Wagner, D.: 2007, Experiences with handheld augmented reality, *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, Washington, DC, USA, pp. 1–13.

Shoemake, K.: 1994, Euler angle conversion, pp. 222–229.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A. and Freeman, W. T.: 2005, Discovering objects and their location in images, *IEEE International Conference on Computer Vision*, Vol. 1, pp. 370–377.

Smit, F. and van Liere, R.: 2008, A framework for performance evaluation of model-based optical trackers, *Eurographics Symposium on Virtual Environments (EGVE)*, Eindhoven, The Netherlands, pp. 33–40.

Stanski, A. and Hellwich, O.: 2005, Spiders as robust point descriptors, *in* W. G. Kropatsch, R. Sablatnig and A. Hanbury (eds), *Pattern Recognition, 27th DAGM Symposium, Vienna, Austria, August 31 - September 2, 2005, Proceedings*, Vol. 3663 of *Lecture Notes in Computer Science*, Springer, pp. 262–268.

Stockham, T.G., J., Cannon, T. and Ingebretsen, R.: 1975, Blind deconvolution through digital signal processing, *Proceedings of the IEEE* **63**(4), 678–692.

Taketa, N., Hayashi, K., Kato, H. and Nishida, S.: 2007, Virtual pop-up book based on augmented reality, *Human Interface and the Management of Information. Interacting in Information Environments*, pp. 475–484.

Tomono, M.: 2007, Monocular slam using a rao-blackwellised particle filter with exhaustive pose space search, *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2421–2426.

Tsai, R.: 1987, A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, *Robotics and Automation, IEEE Journal of* **3**(4), 323–344.

Tuytelaars, T. and Van Gool, L.: 2000, Wide baseline stereo matching based on local, affinely invariant regions, *In Proc. BMVC*, pp. 412–425.

Uematsu, Y. and Saito, H.: 2005, Ar registration by merging multiple planar markers at arbitrary positions and poses via projective space, *ICAT '05: Proceedings of the 2005 international conference on Augmented tele-existence*, ACM, New York, NY, USA, pp. 48–55.

Viola, P. and Jones, M.: 2001, Rapid object detection using a boosted cascade of simple features, *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, pp. 511–518.

Wagner, D. and Schmalstieg, D.: 2003, Artoolkit on the pocketpc platform, *Augmented Reality Toolkit Workshop, 2003. IEEE International* pp. 14–15.

Wagner, D. and Schmalstieg, D.: 2007, Artoolkitplus for pose tracking on mobile devices, *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07), 2007*.

Wagner, M.: 2002, Building wide-area applications with the ar toolkit, *In The First IEEE International Augmented Reality Toolkit Workshop*.

Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K. and Colucci, D.: 1999, The hiball tracker: high-performance wide-area tracking for virtual and augmented environments, *VRST '99: Proceedings of the*

ACM symposium on Virtual reality software and technology, ACM, New York, NY, USA.

Xu, G. and Zhang, Z.: 1996, *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*, Kluwer Academic Publishers, Norwell, MA, USA.

Zhang, X., Fronz, S. and Navab, N.: 2002, Visual marker detection and decoding in ar systems: A comparative study, *ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, Washington, DC, USA, p. 97.

Zhang, Z.: 2000, A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1330–1334.